

Bazy wiedzy

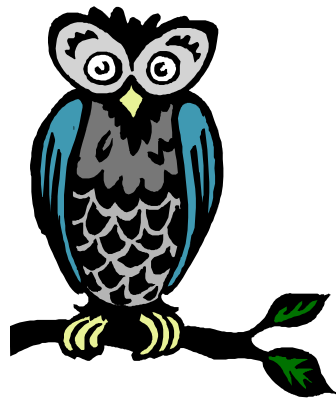
(wykład 4)

prof. dr hab. Krzysztof Goczyła

dr inż. Aleksander Waloszek

dr inż. Wojciech Waloszek

dr inż. Teresa Zawadzka



*Katedra Inżynierii Oprogramowania
Wydział Elektroniki, Telekomunikacji
i Informatyki
Politechnika Gdańska*

Modelowanie

Kształtowanie dziedziny

Modelowanie możemy traktować jako proces kształtowania dziedziny. Proces ten nazywa się także *konceptualizacją*.

1. Czym jest dziedzina?
2. Jakie mamy narzędzia?
3. Jakimi zasadami powinniśmy się kierować?

Modelowanie

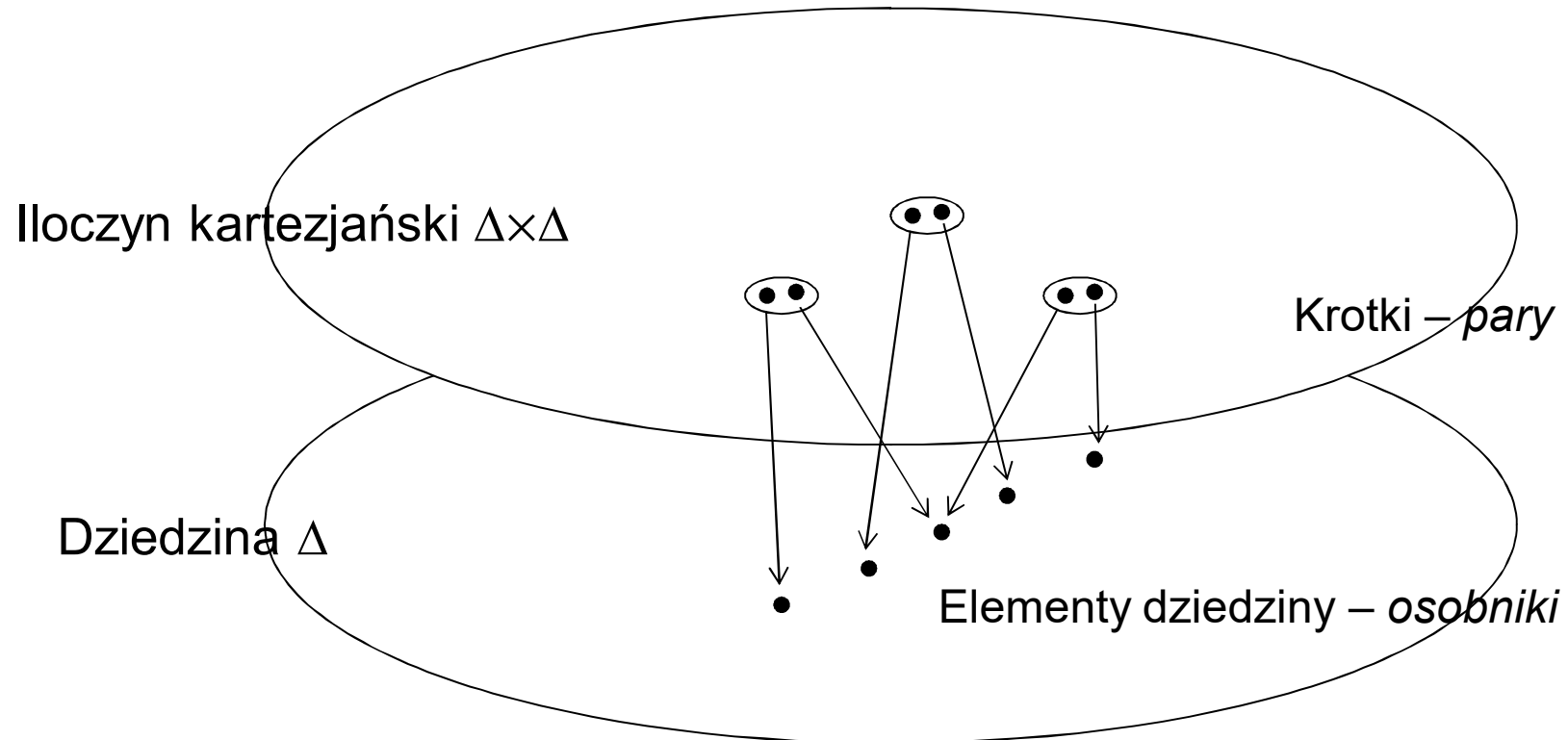
Czym jest dziedzina?

Dziedzina Δ



Modelowanie

Czym jest dziedzina?



Modelowanie

Czym jest dziedzina?

Iloczyn kartezjański $\Delta \times \Delta \times \Delta$

Krotki – *trójki*

Iloczyn kartezjański $\Delta \times \Delta$

Krotki – *pary*

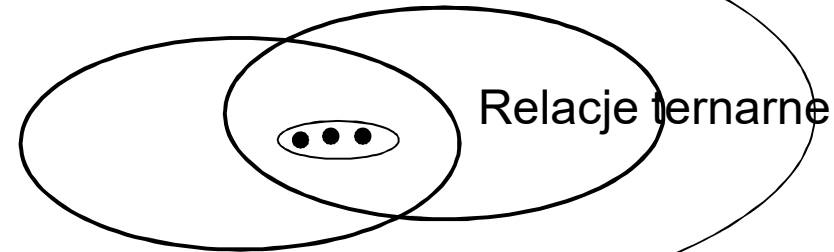
Dziedzina Δ

Elementy dziedziny – *osobniki*

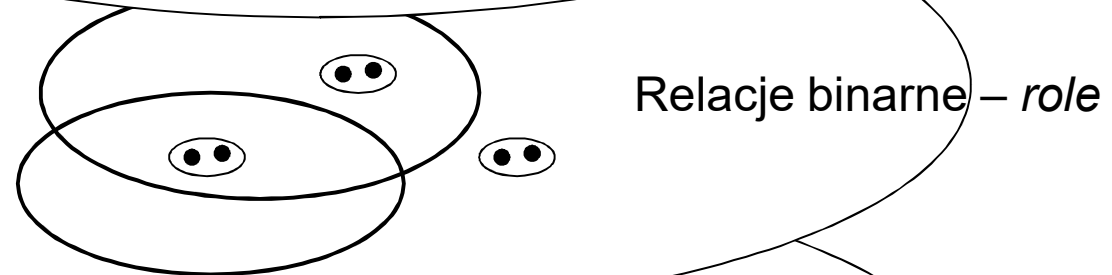
Modelowanie

Czym jest dziedzina?

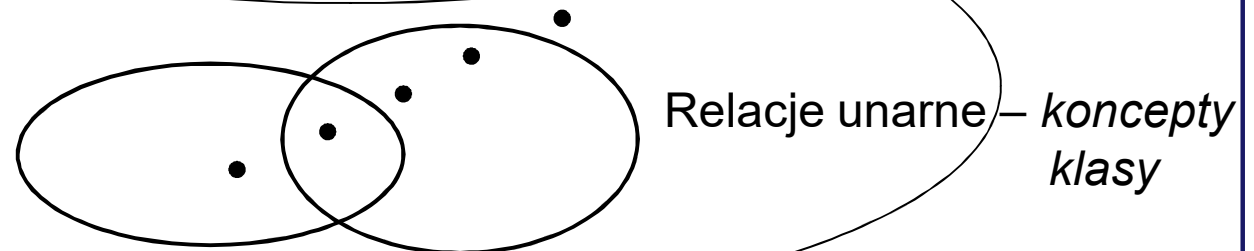
Iloczyn kartezjański $\Delta \times \Delta \times \Delta$



Iloczyn kartezjański $\Delta \times \Delta$



Dziedzina Δ



Modelowanie

Jakie mamy narzędzia?

Narzędzie 1: co znajduje się w dziedzinie?

Willard van Orman Quine:

Ontologia nie odpowiada na pytanie „co istnieje?”, a jedynie na pytanie, jakie są *zobowiązania ontologiczne* danej **teorii aksjomatycznej**.

Co powinniśmy uznać za istniejące, gdy uznajemy tę teorię?

„To be is to be the value of a variable”

– Quine: *On What There Is*, Review of Metaphysics 1948

Inaczej mówiąc, co podlega kwantyfikacji (\exists , \forall)?

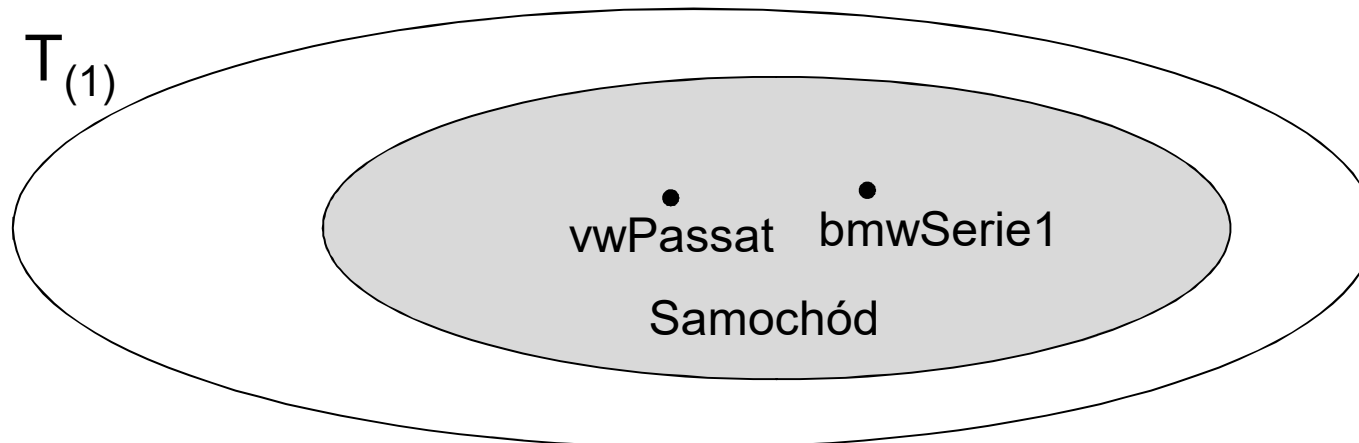
Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

Samochód(vwPassat)
Samochód(bmwSerie1)

Wystąpieniami konceptu Samochód są *de facto* modele samochodów. Ale bardzo często abstrahujemy od pojedynczych egzemplarzy, nazywając modele samochodami. Np. na pytanie: „jakie znasz samochody produkcji niemieckiej?” większość ludzi wymieni modele, a nie egzemplarze.



Modelowanie

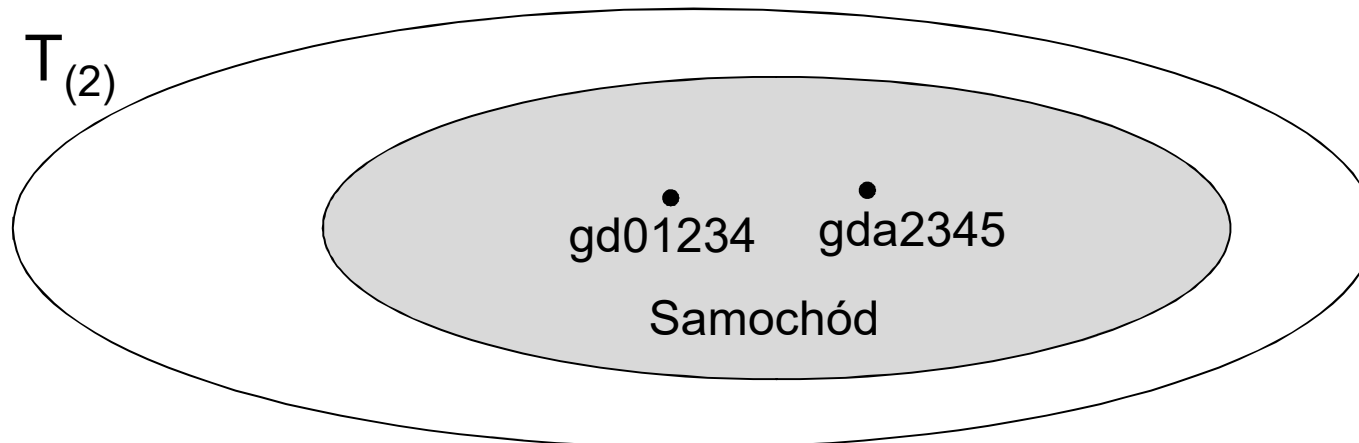
Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

Samochód(gd01234)

Samochód(gda2345)

Tutaj wystąpieniami konceptu Samochód są już egzemplarze. Ponieważ poprzednio abstrahowaliśmy od egzemplarzy, to $T_{(1)} \neq T_{(2)}$ (pomimo że obie dziedziny zawierają koncept Samochód).



Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

Model \sqcap Samochód $\sqsubseteq \perp$

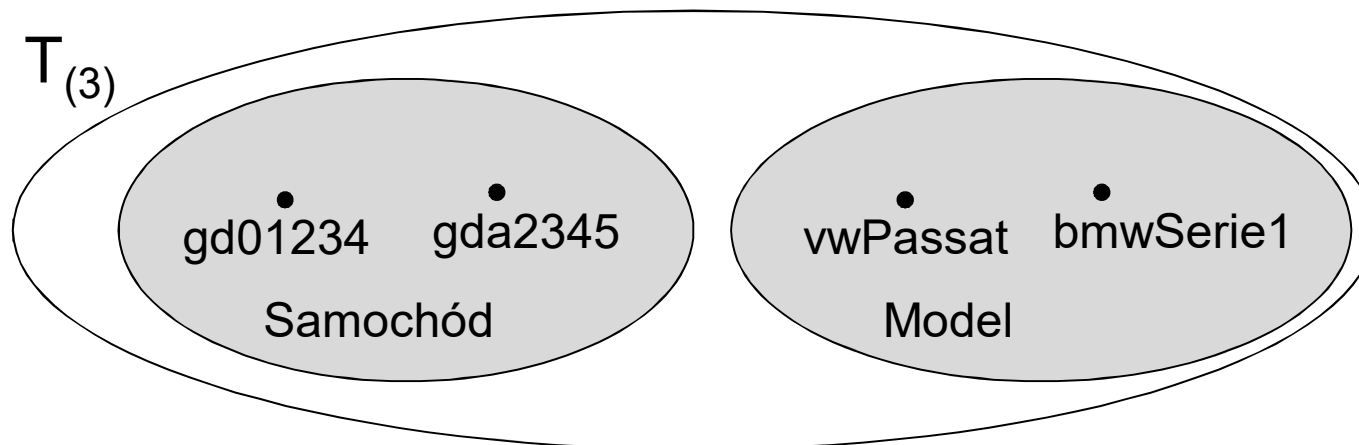
Samochód(gd01234)

Samochód(gda2345)

Model(vwPassat)

Model(bmwSerie1)

Możemy powiązać obie dziedziny, ale nie będzie to operacja prosta: musimy zmienić nazwę konceptu Samochód z dziedziny T(1) na Model oraz dodać aksjomat o rozłączności konceptów Model i Samochód.



Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

Model \sqcap Samochód $\sqsubseteq \perp$

Samochód \sqsubseteq

\exists jestEgzemplarzem.Model

Samochód(gd01234)

Samochód(gda2345)

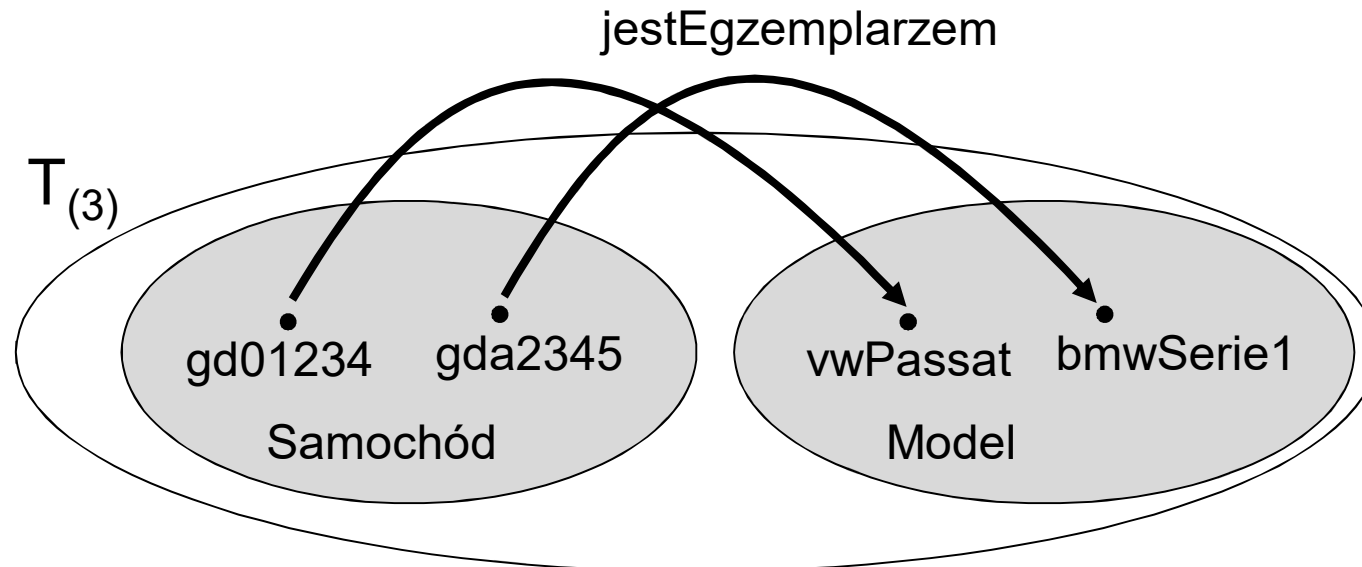
Model(vwPassat)

Model(bmwSerie1)

jestEgzemplarzem(gd01234, vwPassat)

jestEgzemplarzem(gda2345, bmwSerie1)

Model staje się przydzielonym za pomocą roli jestEgzemplarzem atrybutem typu obiektowego. (O atrybutach będziemy jeszcze mówić)



Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

Model \sqcap Samochód $\sqsubseteq \perp$

Samochód \sqsubseteq

\exists jestEgzemplarzem.Model

Na ERD zamodelowalibyśmy tak:



Zwróćmy uwagę: gd01234 nie jest tutaj numerem rejestracyjnym, a vwPassat nazwą modelu. Z punktu widzenia logiki są to nazwy zmiennych, a z punktu widzenia ERD możemy je potraktować co najwyżej jak (sztuczne) identyfikatory.

Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

Model \sqcap Samochód $\sqsubseteq \perp$

Samochód \sqsubseteq

\exists jestEgzemplarzem.Model

Samochód(gd01234)

Samochód(gda2345)

Model(vwPassat)

Model(bmwSerie1)

jestEgzemplarzem(gd01234, vwPassat)

jestEgzemplarzem(gda2345, bmwSerie1)

A w bazie danych byłoby tak:

Tabela Samochody

Id	FK_model
gd01234	vwPassat
gda2345	bmwSerie1

Tabela Modele

Id
vwPassat
bmwSerie1

----->

----->

Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

VwPassat \sqsubseteq Samochód

BmwSerie1 \sqsubseteq Samochód

VwPassat(gd01234)

BmwSerie1(gda2345)

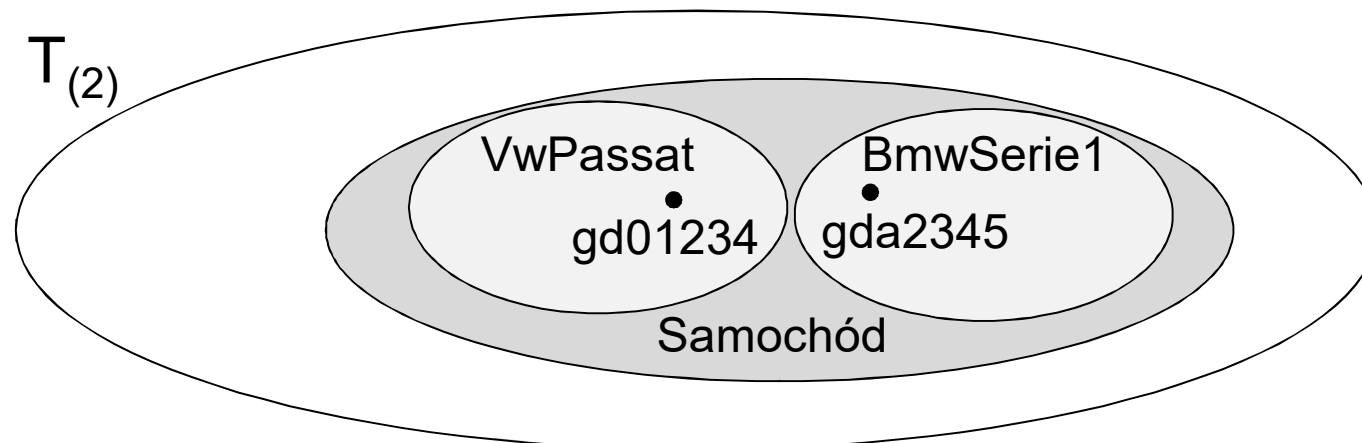
Ale wróćmy do dziedziny $T_{(2)}$ (tej z egzemplarzami samochodów i bez modeli). Przynależność egzemplarzy do modeli możemy też wyrazić, definiując modele jako subkoncepty konceptu Samochód.

Takie ujęcie lepiej oddaje rzeczywistość. Można np.

-model jest prostszy, wnioskowanie mniej złożone,

-łatwo wyrazić hierarchię, np. Golf \sqsubseteq VW, Sportvan \sqsubseteq Golf itp.

-łatwo wyrazić rozłączność, wyczerpywanie itp.



Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

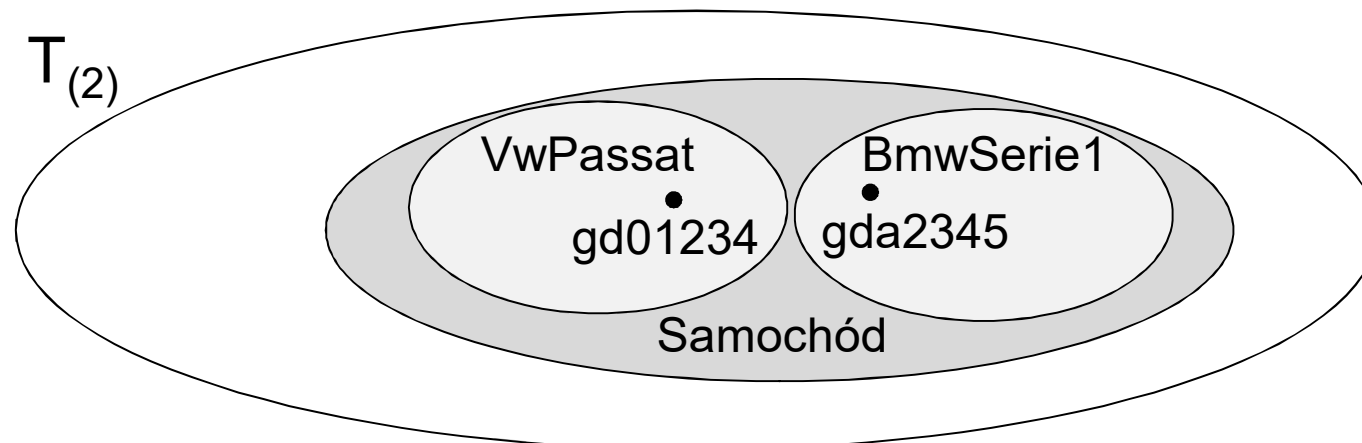
VwPassat \sqsubseteq Samochód

BmwSerie1 \sqsubseteq Samochód

VwPassat(gd01234)

BmwSerie1(gda2345)

Ma też swoje wady: wiedzę o modelach umieszczamy na poziomie schematu, więc po pojawieniu się nowego modelu musimy zmienić schemat bazy.



Modelowanie

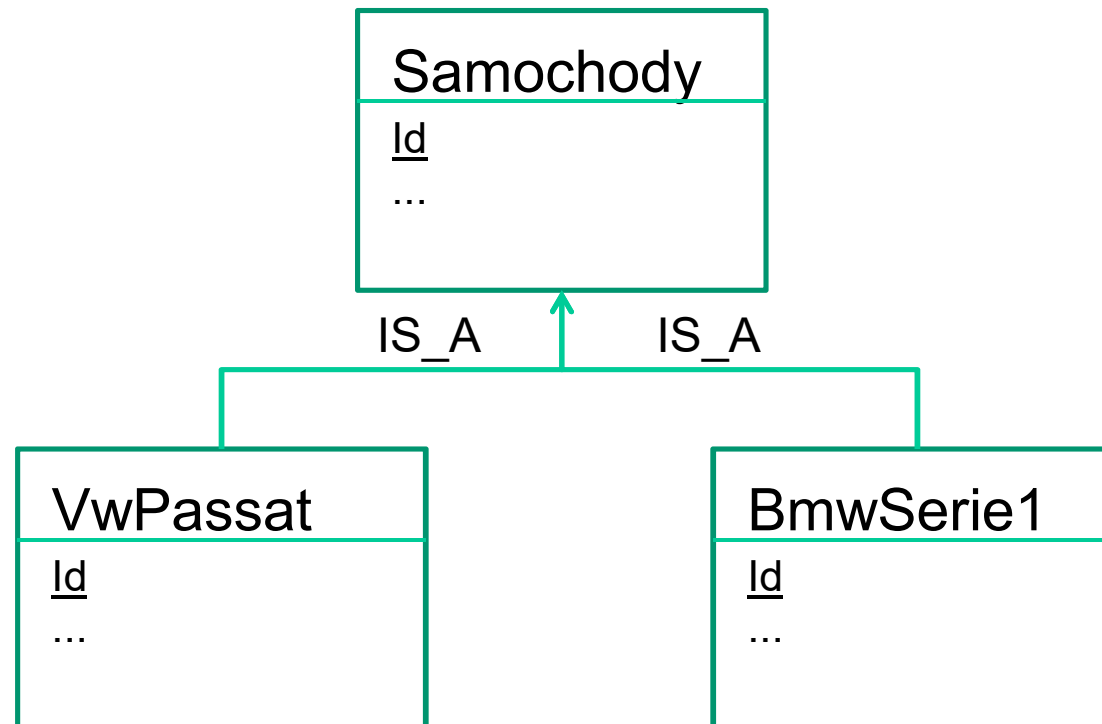
Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

VwPassat \sqsubseteq Samochód

BmwSerie1 \sqsubseteq Samochód

Na ERD:



Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

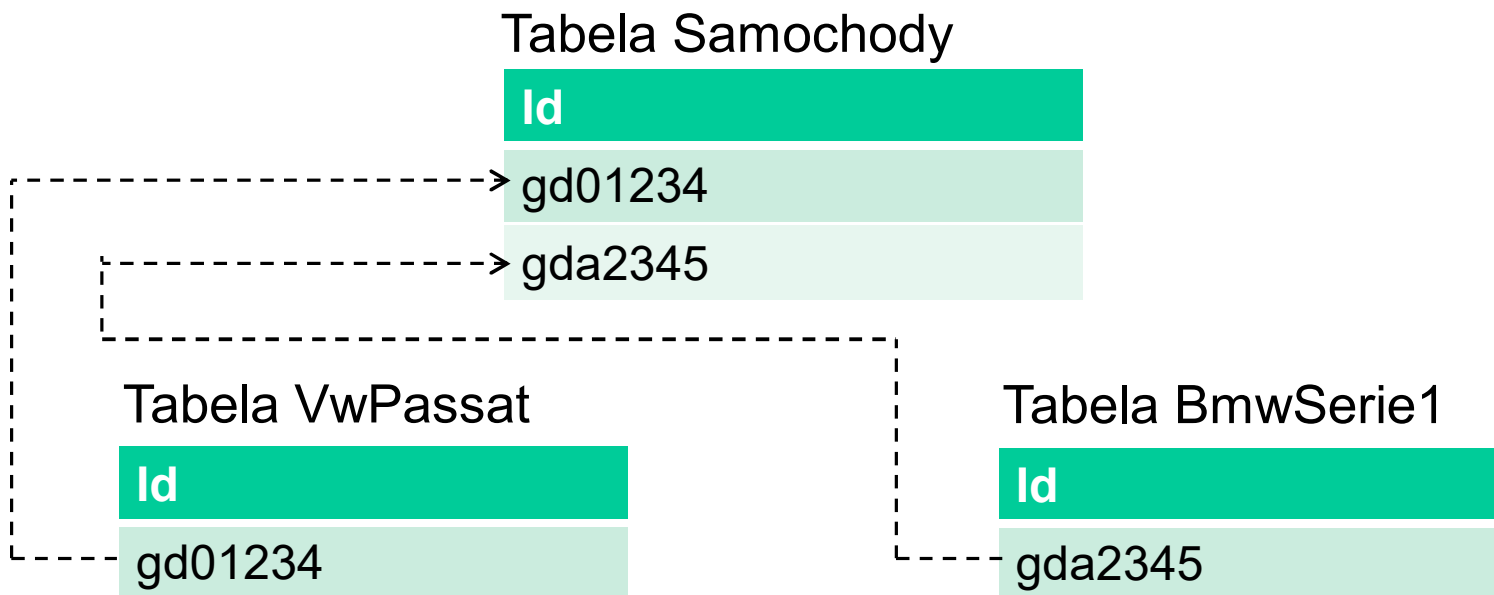
VwPassat \sqsubseteq Samochód

BmwSerie1 \sqsubseteq Samochód

VwPassat(gd01234)

BmwSerie1(gda2345)

A bazie danych:



Modelowanie

Narzędzie 1: co zawiera dziedzina?

Samochód $\sqsubseteq \top$

VwPassat \sqsubseteq Samochód

BmwSerie1 \sqsubseteq Samochód

VwPassat(gd01234)

BmwSerie1(gda2345)

W praktyce często implementujemy dziedziczenie w jednej tabeli. Wtedy trochę zbliżamy się do modelu z dziedziny $T_{(3)}$

Ale tutaj jest gorzej: zamieniamy przecież concept na wartość atrybutu (łańcuch znakowy: „VwPassat”), czyli symbol, a nie na encję. Nie będziemy więc mogli wypowiadać się na temat wzajemnych związków między modelami oraz związków między modelami a innymi encjami.

Tabela Samochody

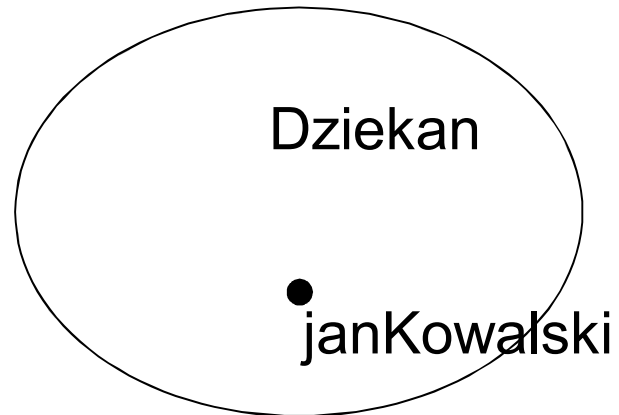
Id	Typ
gd01234	VwPassat
gda2345	BmwSerie1

Modelowanie

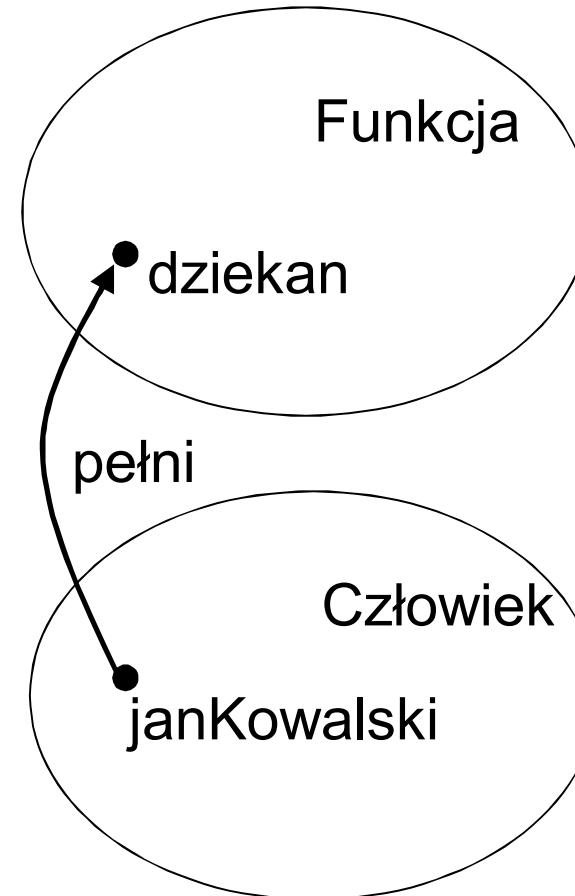
Narzędzie 2: koncepty vs role

Jan Kowalski jest dziekanem.

Dziekan(janKowalski)



pełni(janKowalski, dziekan)

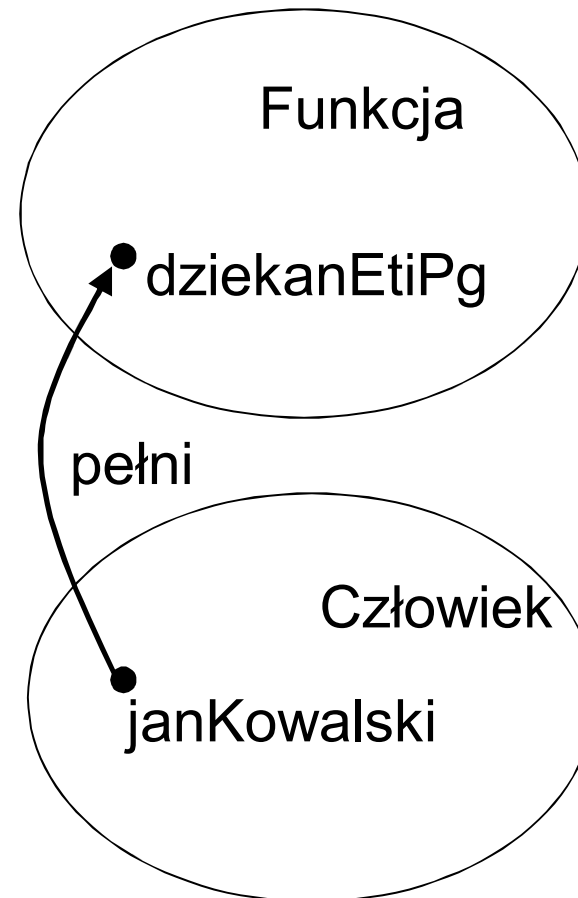
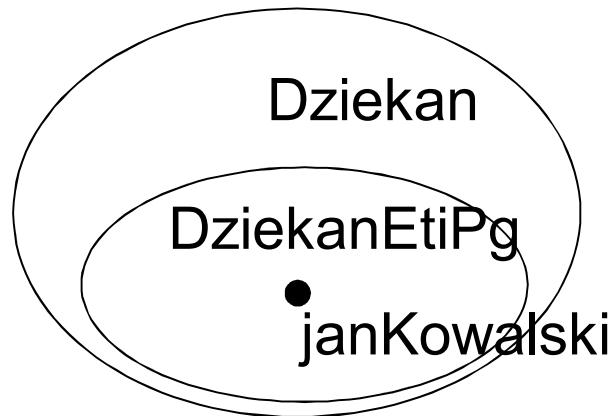


Modelowanie

Narzędzie 2: koncepty vs role

Jan Kowalski jest dziekanem ETI PG.

DziekanEtiPg(janKowalski) pełni(janKowalski, dziekanEtiPg)



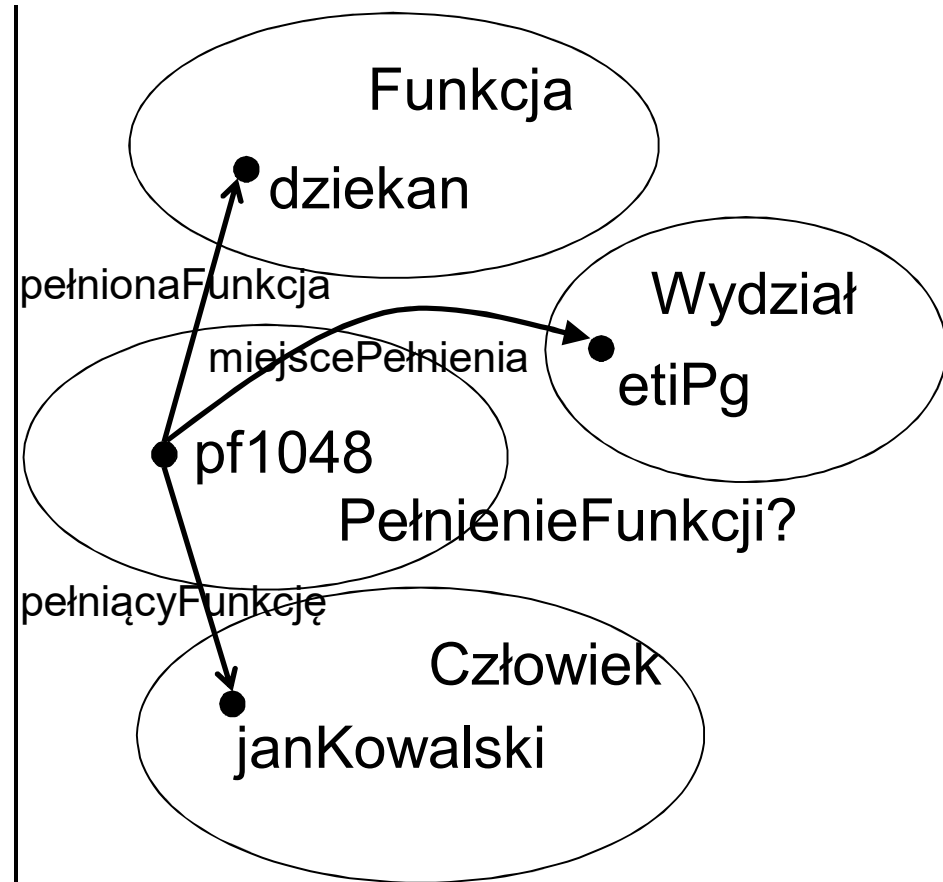
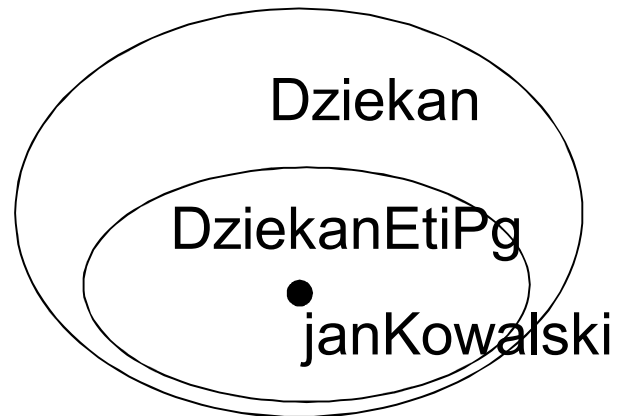
Ale jak powiedzieć, że dziekanEtiPg jest dziekanem?

Modelowanie

Narzędzie 2: koncepty vs role

Jan Kowalski jest dziekanem ETI PG.

DziekanEtiPg(janKowalski) Wiele asercji...



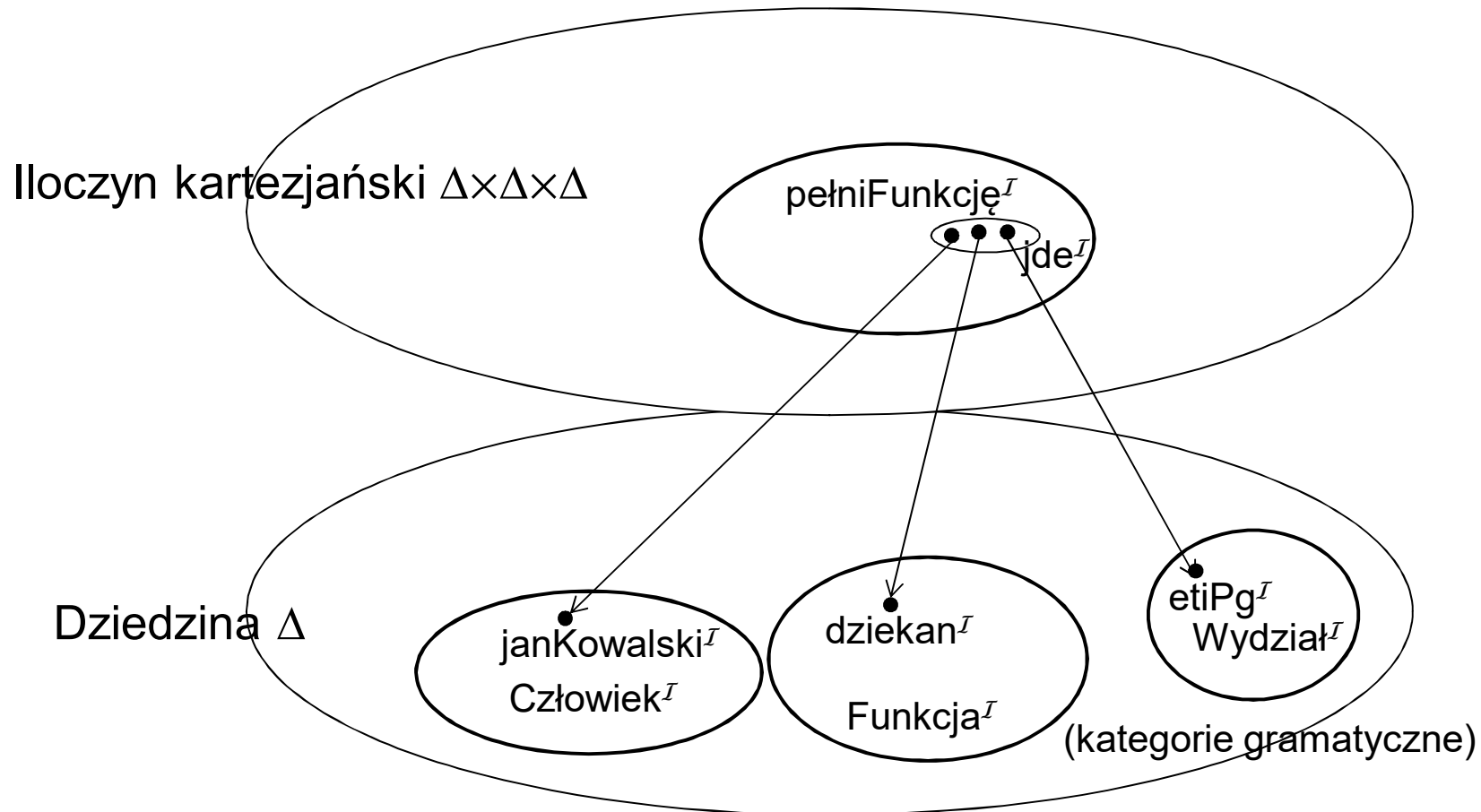
Reifikacja relacji

Modelowanie

Narzędzie 2: koncepty vs role

Jan Kowalski jest dziekanem ETI PG.

$\text{pełniFunkcję}(\text{janKowalski}, \text{dziekan}, \text{etiPg})$ – predykat 3-go stopnia



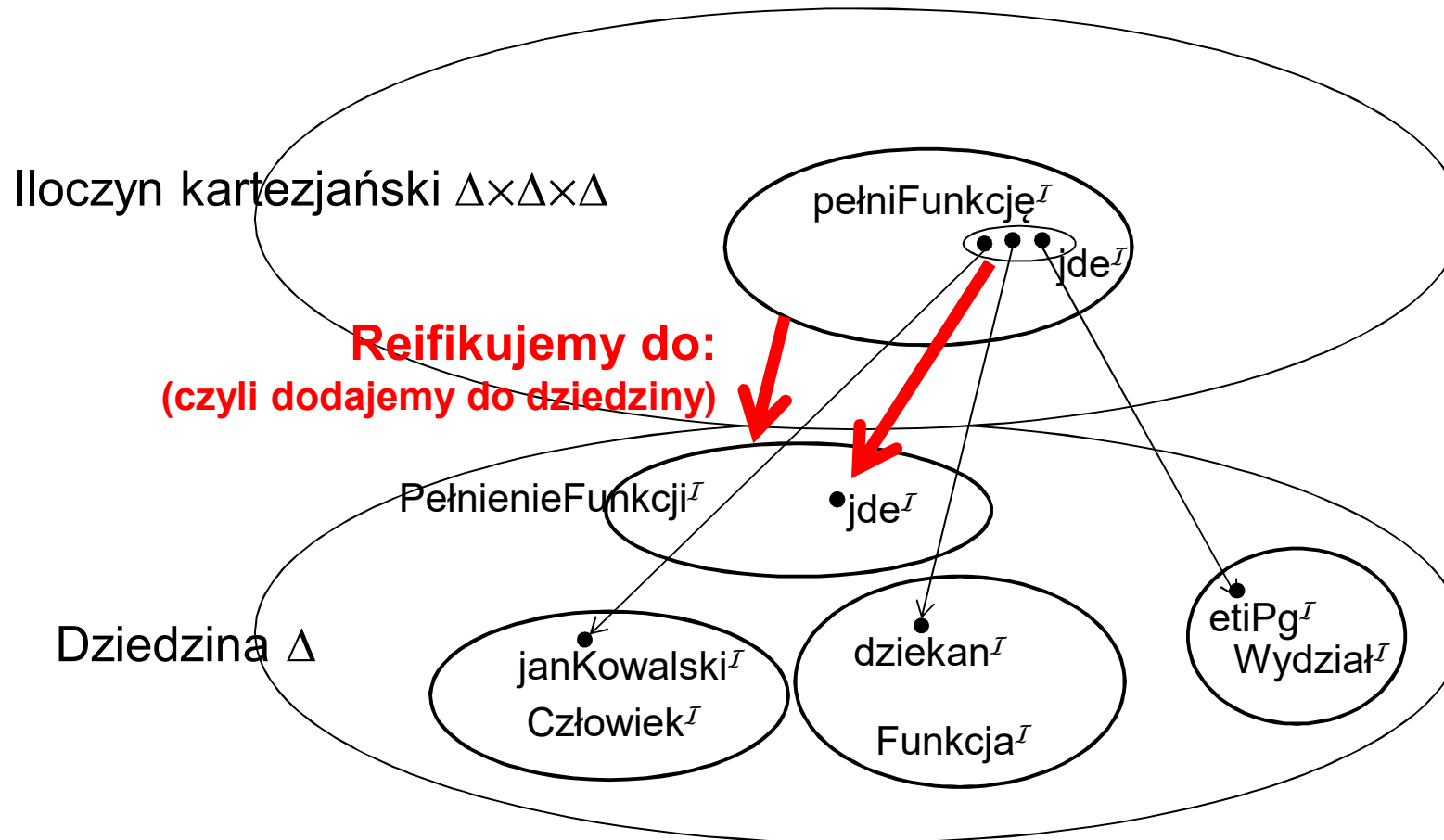
Dlaczego reifikacja relacji?

Modelowanie

Narzędzie 2: koncepty vs role

Jan Kowalski jest dziekanem ETI PG.

$\text{pełniFunkcję}(\text{janKowalski}, \text{dziekan}, \text{etiPg})$ – predykat 3-go stopnia



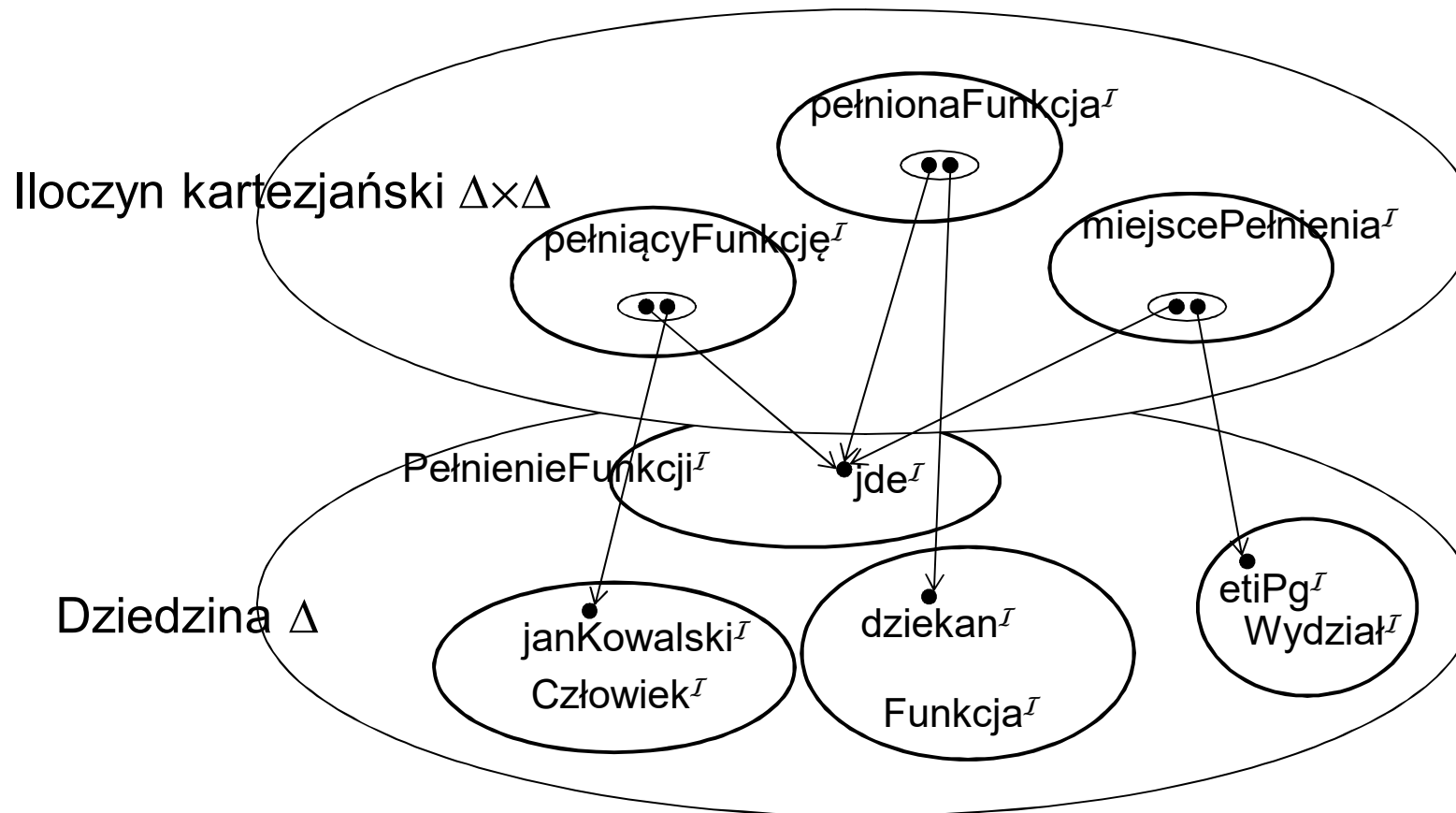
Dlaczego reifikacja relacji?

Modelowanie

Narzędzie 2: koncepty vs role

Jan Kowalski jest dziekanem ETI PG.

$\text{pełniFunkcję}(\text{janKowalski}, \text{dziekan}, \text{etiPg})$ – predykat 3-go stopnia

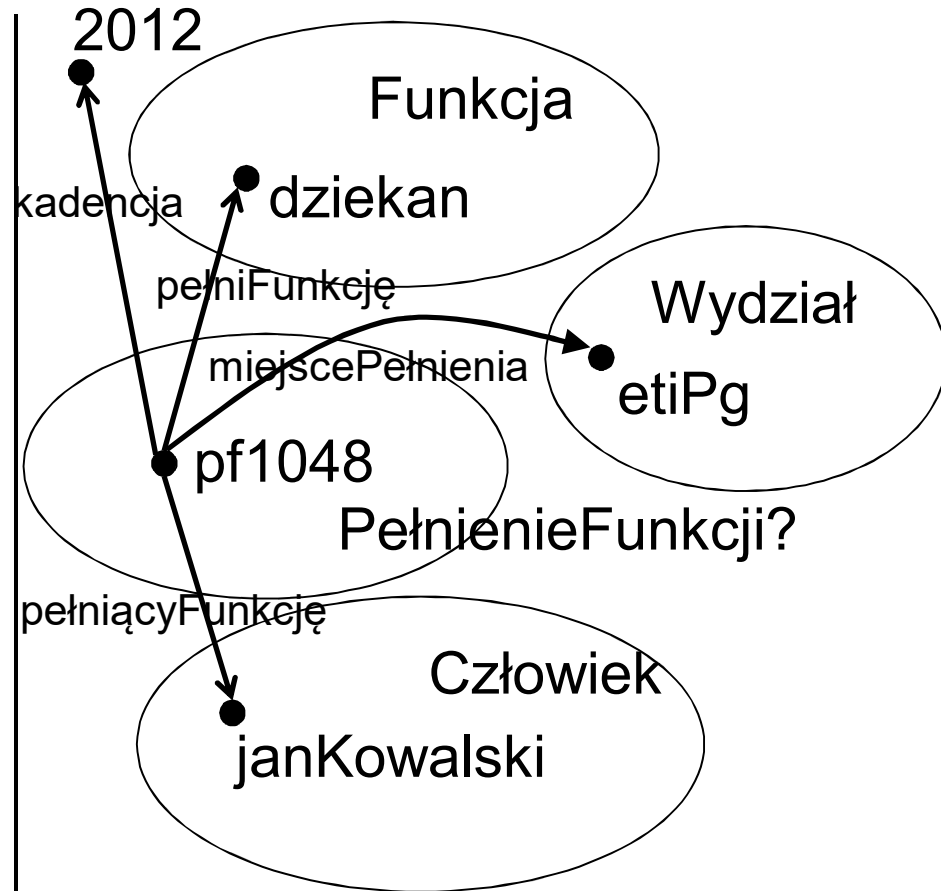
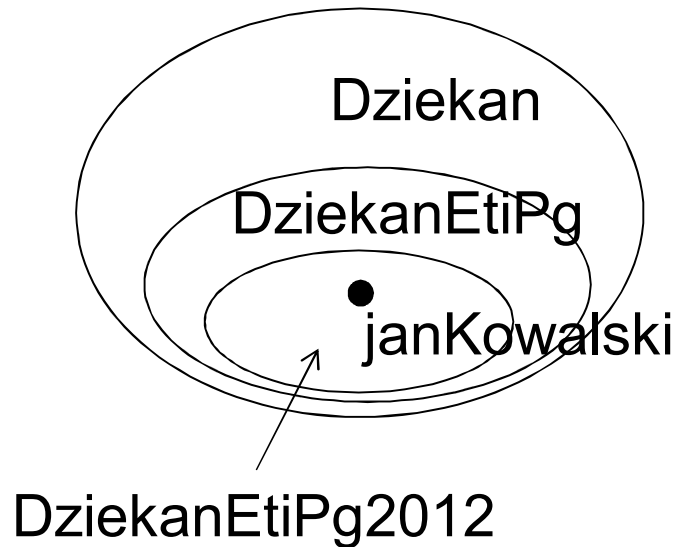


Reifikacja relacji

Modelowanie

Narzędzie 2: koncepty vs role

Jan Kowalski jest dziekanem ETI PG kadencji 2012.
 DziekanEtiPg(janKowalski) Wiele asercji...

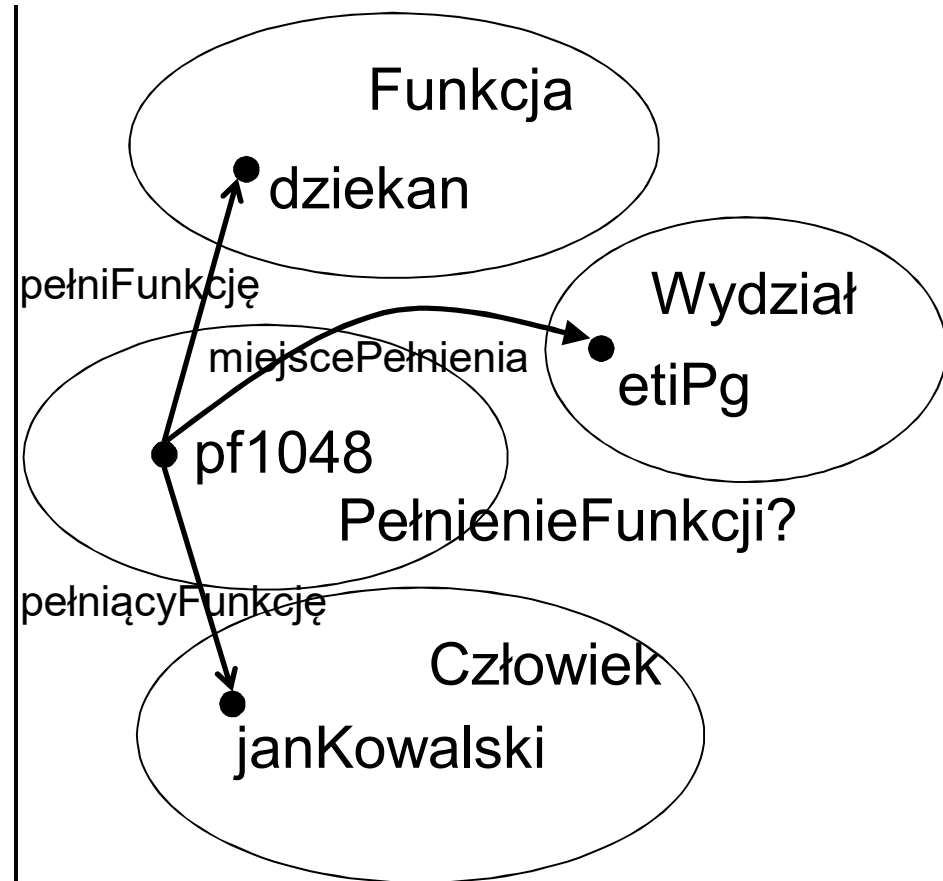
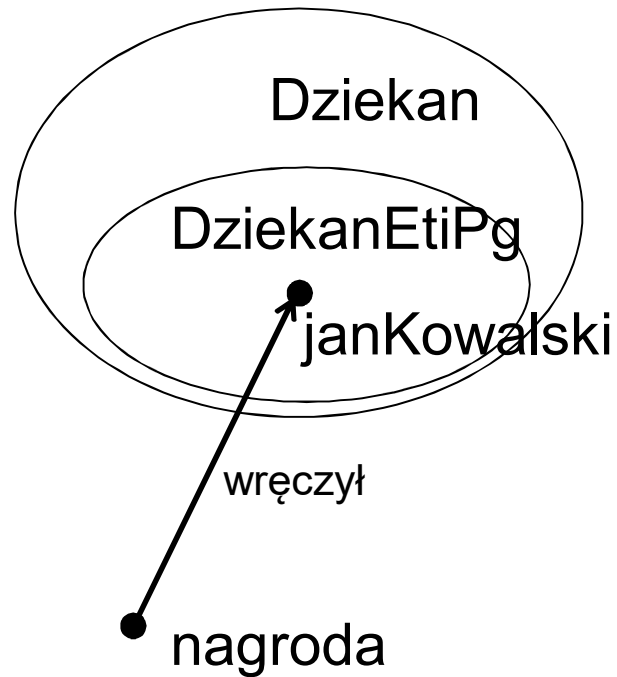


Reifikacja relacji

Modelowanie

Narzędzie 2: koncepty vs role

Dziekan Jan Kowalski wręczył nagrodę.
 DziekanEtiPg(janKowalski) Wiele asercji...

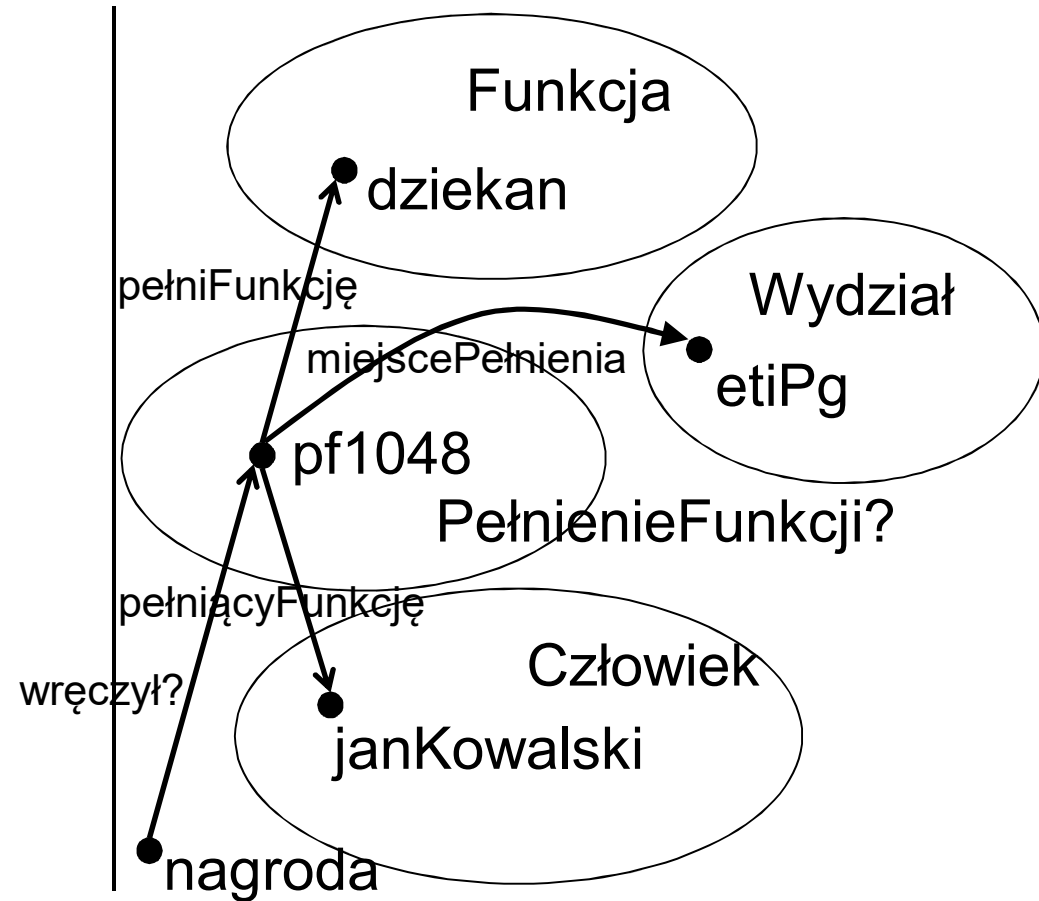
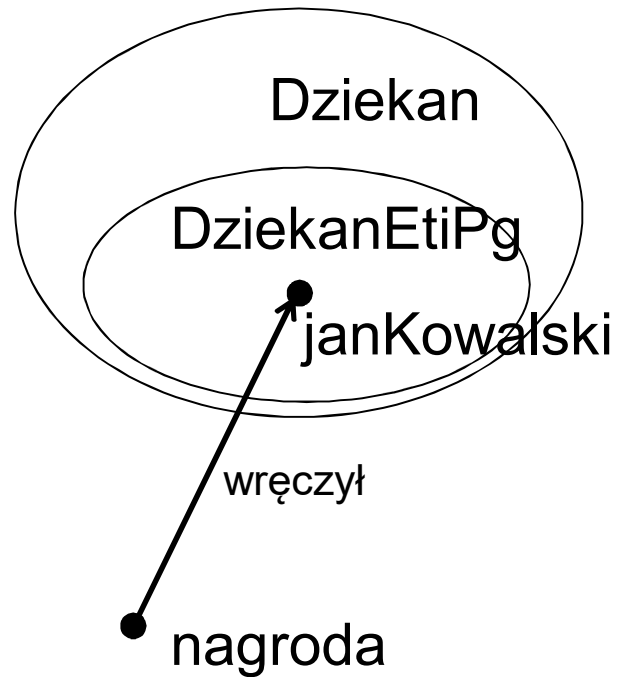


Reifikacja relacji

Modelowanie

Narzędzie 2: koncepty vs role

Dziekan Jan Kowalski wręczył nagrodę.
 DziekanEtiPg(janKowalski) Wiele asercji...

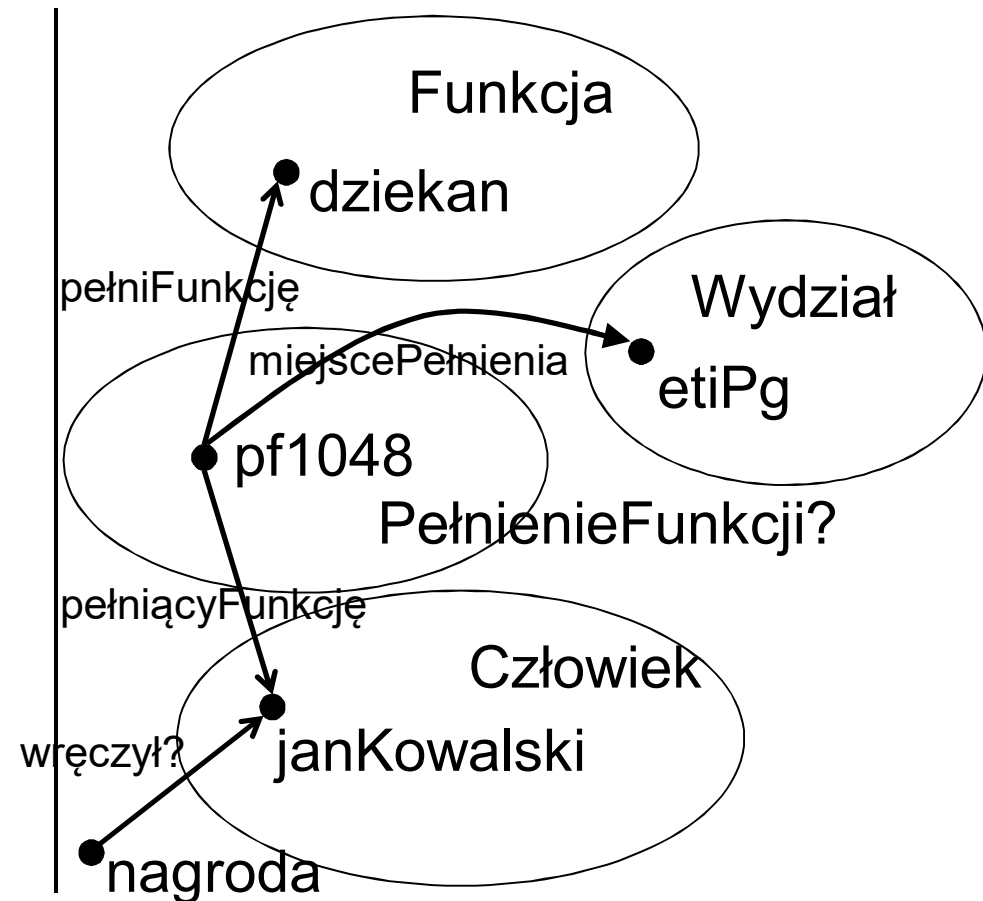
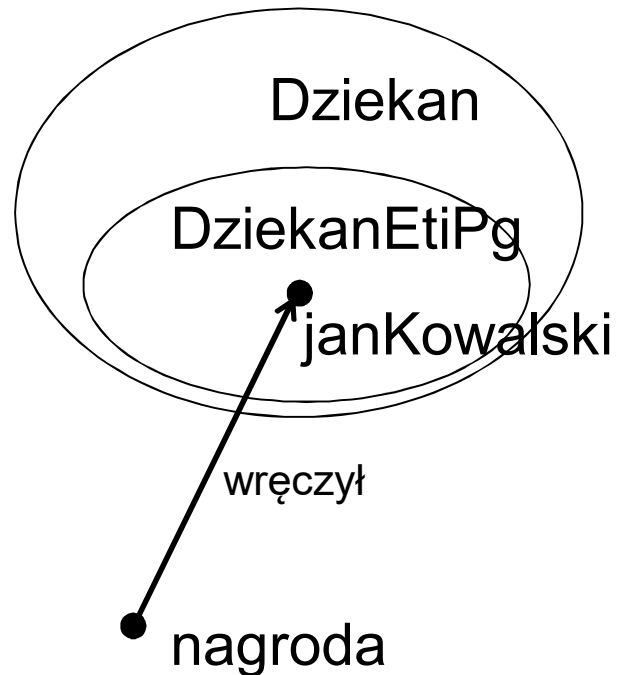


Reifikacja relacji

Modelowanie

Narzędzie 2: koncepty vs role

Dziekan Jan Kowalski wręczył nagrodę.
DziekanEtiPg(janKowalski) Wiele asercji...



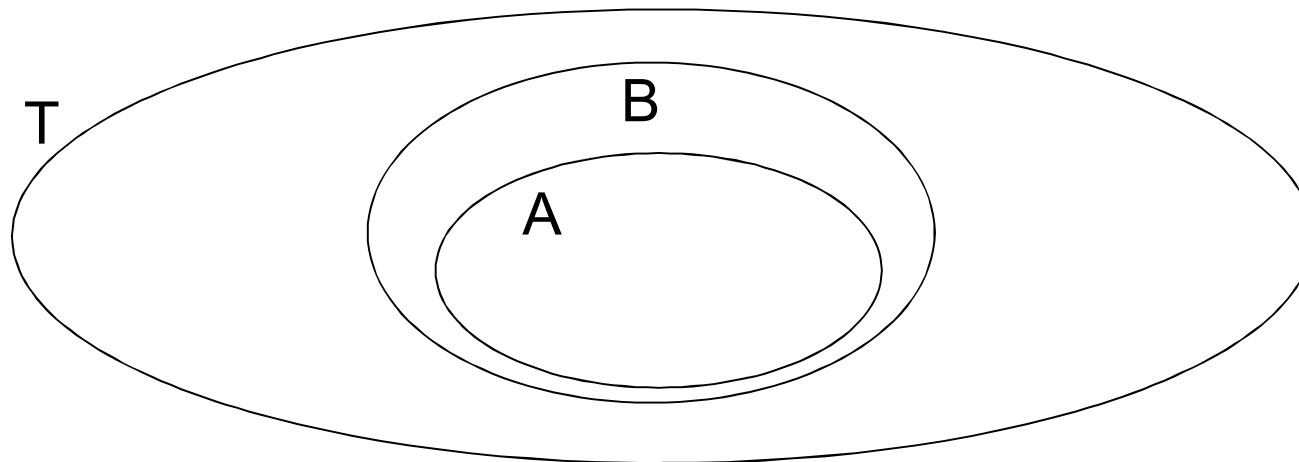
Reifikacja relacji

Modelowanie

Narzędzie 3: dziedziczenie

Dziedziczenie, czyli subsumcja (ang. *subsumption*)
A dziedziczy od B (dziedziczenie B przez A)
B subsumuje A (subsumcja A przez B)

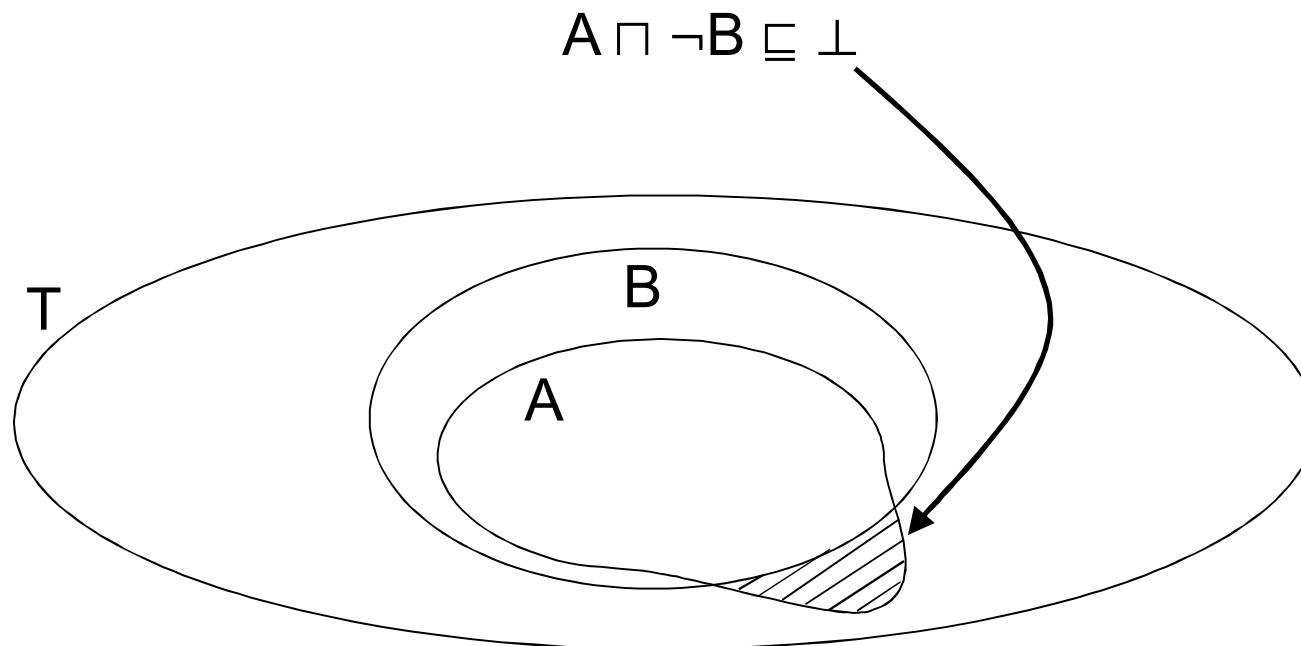
$$A \sqsubseteq B$$



Modelowanie

Narzędzie 3: dziedziczenie

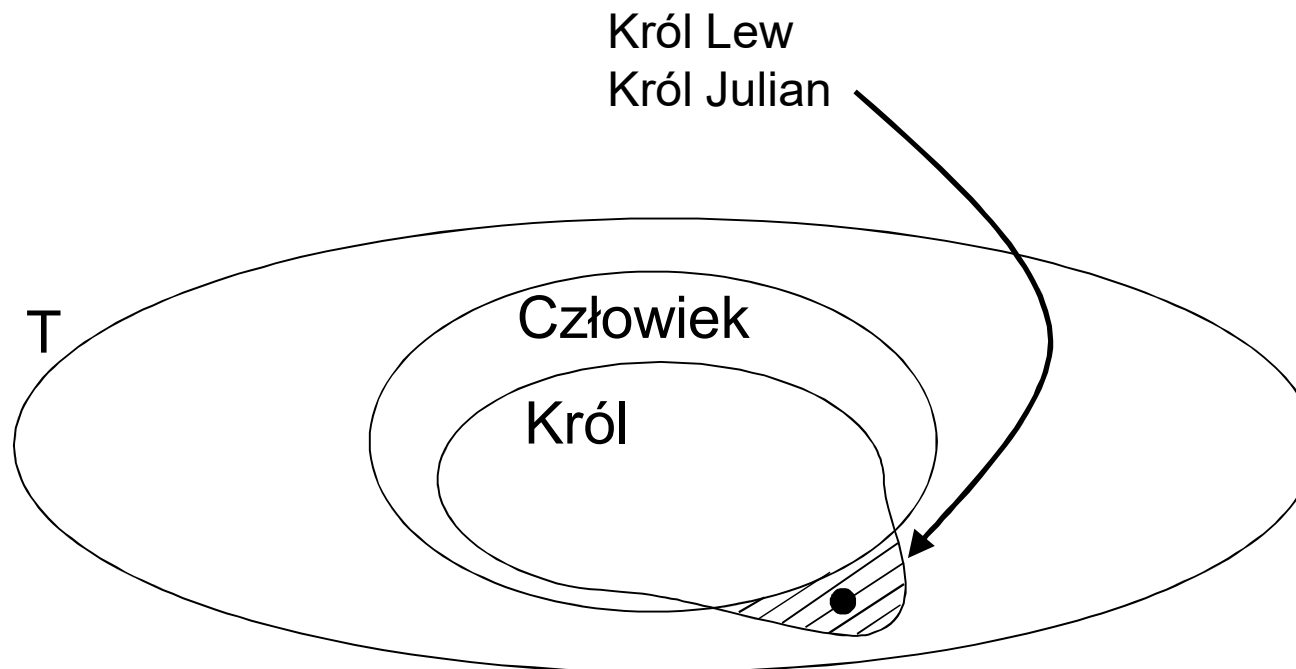
Oznacza, że usuwamy z dziedziny przypadki istnienia elementów dziedziny będących A, a nie będących B!



Modelowanie

Narzędzie 3: dziedziczenie

Oznacza, że usuwamy z dziedziny przypadki istnienia elementów dziedziny będących A, a nie będących B!

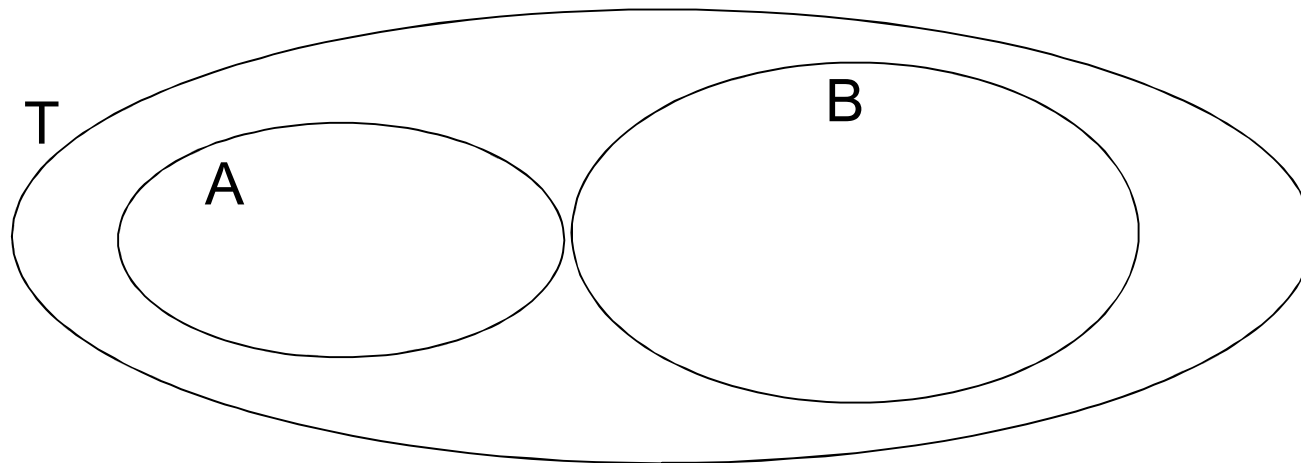


Modelowanie

Narzędzie 3: dziedziczenie

Rozłączność (ang. *disjointness*)
A rozłączne z B

$$A \cap B \subseteq \perp$$

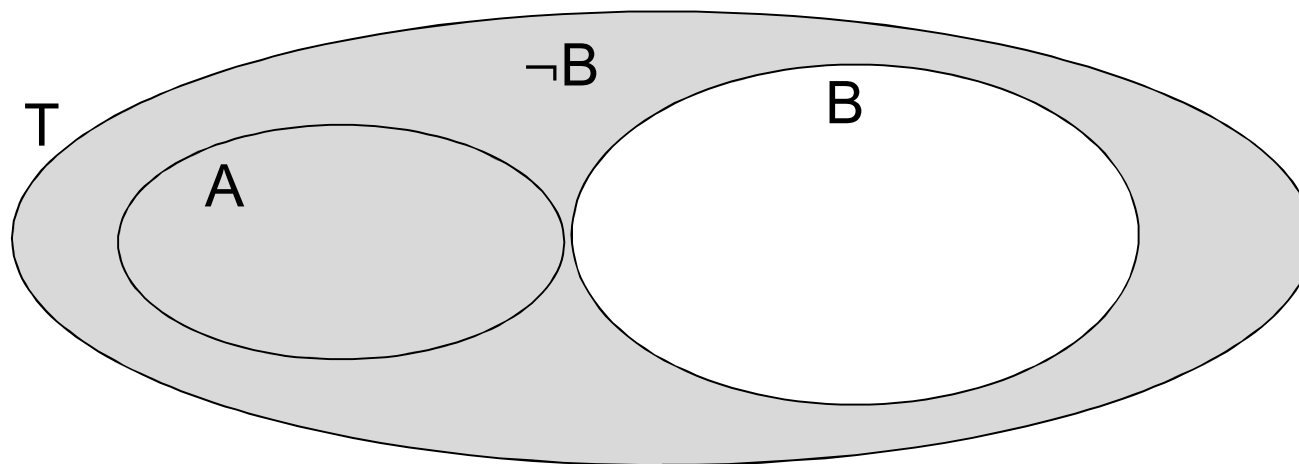


Modelowanie

Narzędzie 3: dziedziczenie

Rozłączność da się sprowadzić do dziedziczenia:

$$A \subseteq \neg B$$

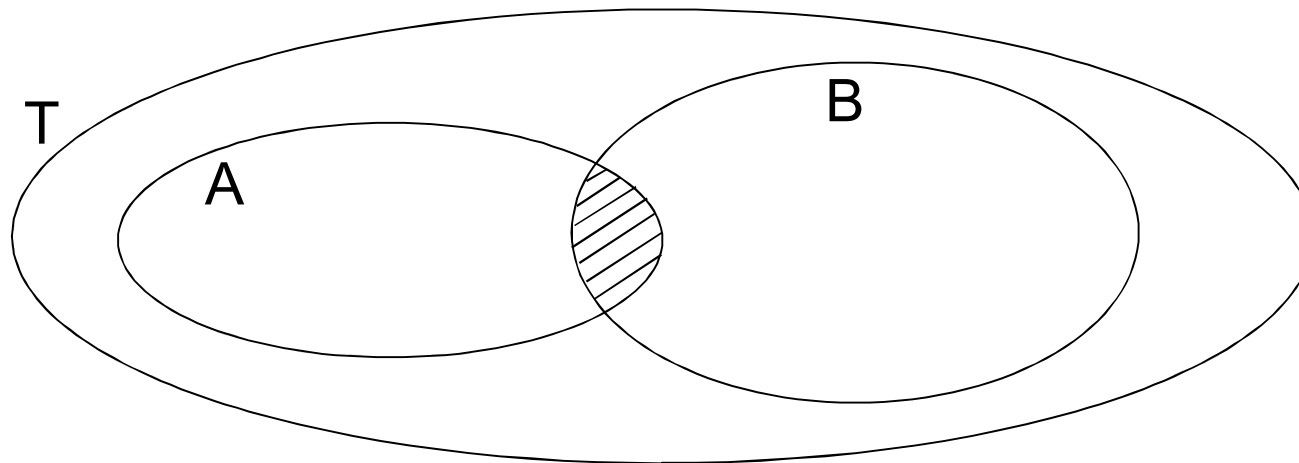


Modelowanie

Narzędzie 3: dziedziczenie

Oznacza, że usuwamy z dziedziny przypadki istnienia elementów dziedziny będących A i będących B!

$$A \cap B \subseteq \perp$$

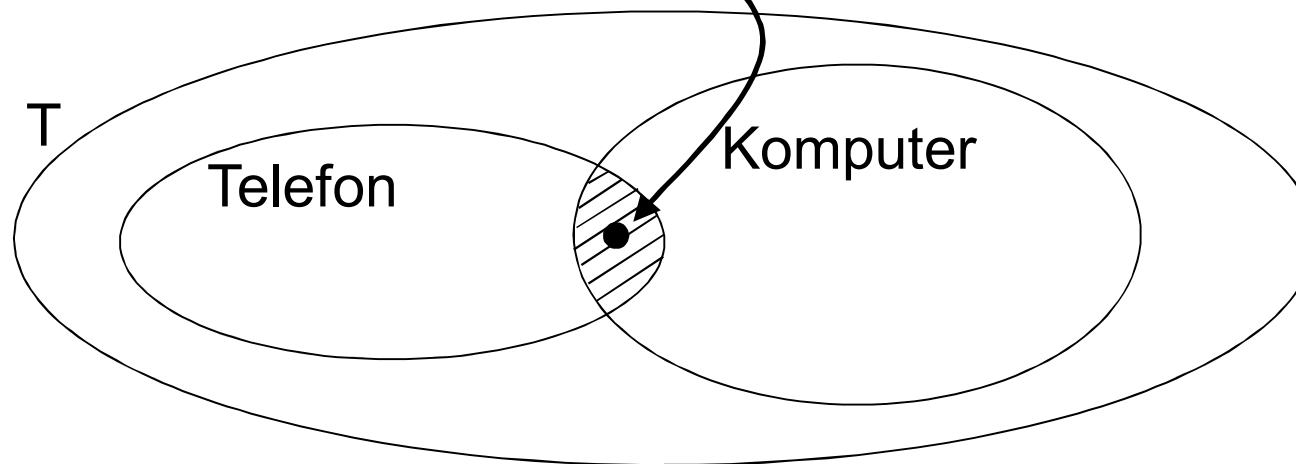


Modelowanie

Narzędzie 3: dziedziczenie

Oznacza, że usuwamy z dziedziny przypadki istnienia elementów dziedziny będących A i będących B!

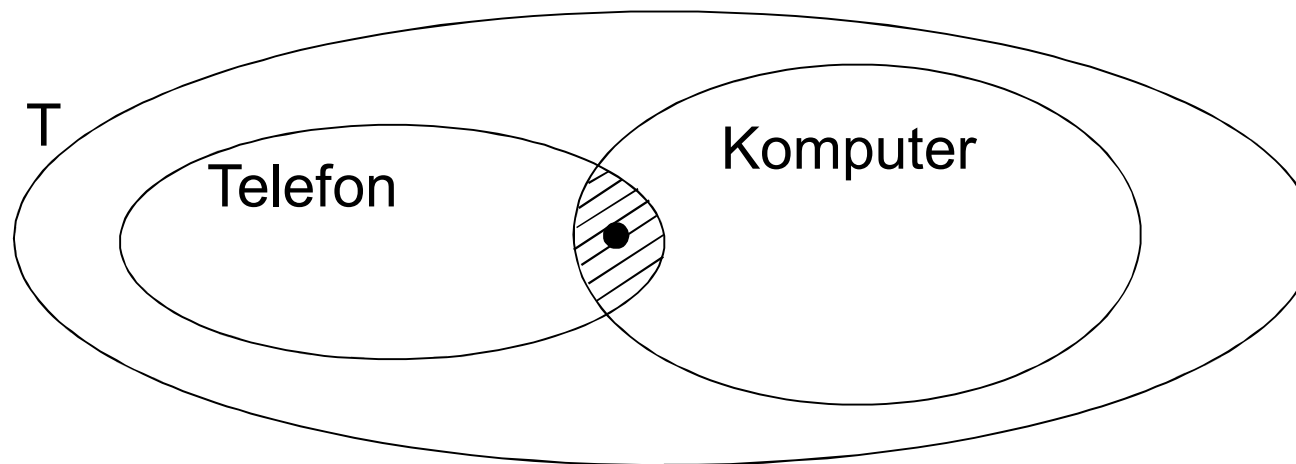
Telefonokomputer (smartfon)



Modelowanie

Narzędzie 3: dziedziczenie

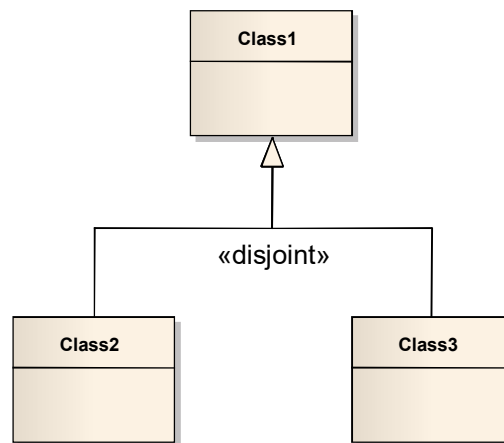
Oznacza, że usuwamy z dziedziny przypadki istnienia elementów dziedziny będących A i będących B!



Modelowanie

Narzędzie 3: dziedziczenie

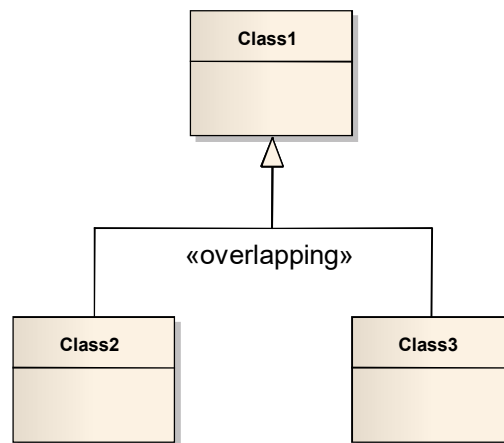
W UML-u jest możliwe zadeklarowanie rozłączności klas, ale tylko dziedziczących od jakiejś klasy trzeciej (Class1). Właściwie, to nie wiadomo po co, gdyż w modelu obiektowym nie ma możliwości stworzyć obiektu należący do kilku klas (niedziedziczących). Rozłączność jest więc niejako domyślna.



Modelowanie

Narzędzie 3: dziedziczenie

Ale istnieje też możliwość zdefiniowania klas, które nie są rozłączne.

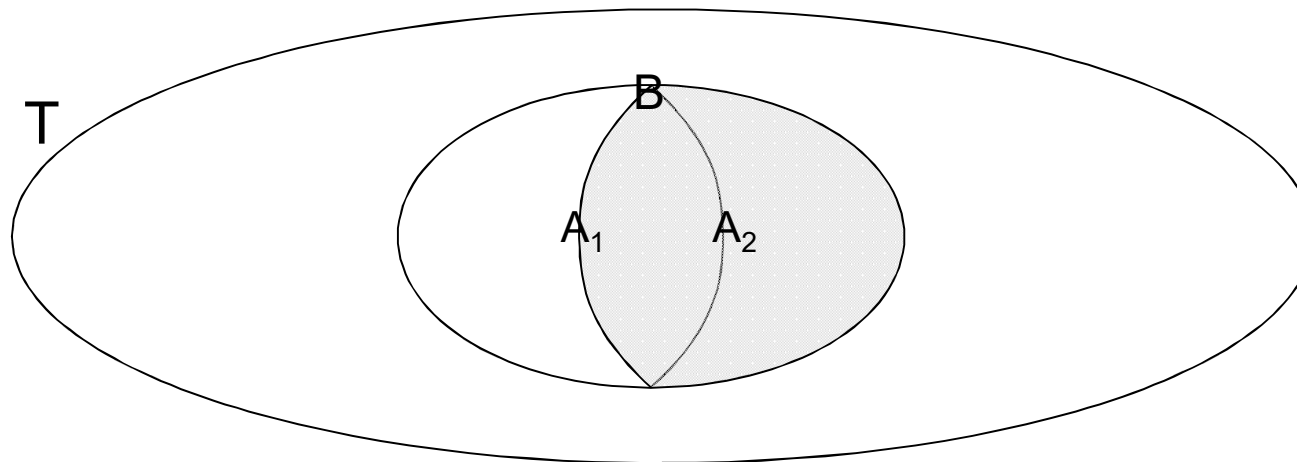


Modelowanie

Narzędzie 3: dziedziczenie

Dziedziczenie kompletne (ang. complete inheritance) oznacza, że każde wystąpienie konceptu dziedziczonego jest także wystąpieniem któregoś z konceptów dziedziczących

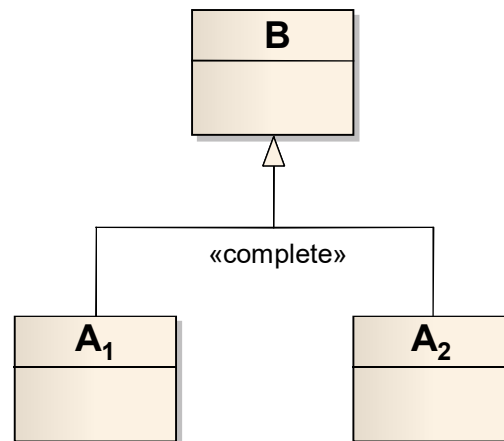
$$A_1 \sqcup A_2 \equiv B$$



Modelowanie

Narzędzie 3: dziedziczenie

W UML-u jest możliwe zadeklarowanie kompletności dziedziczenia.

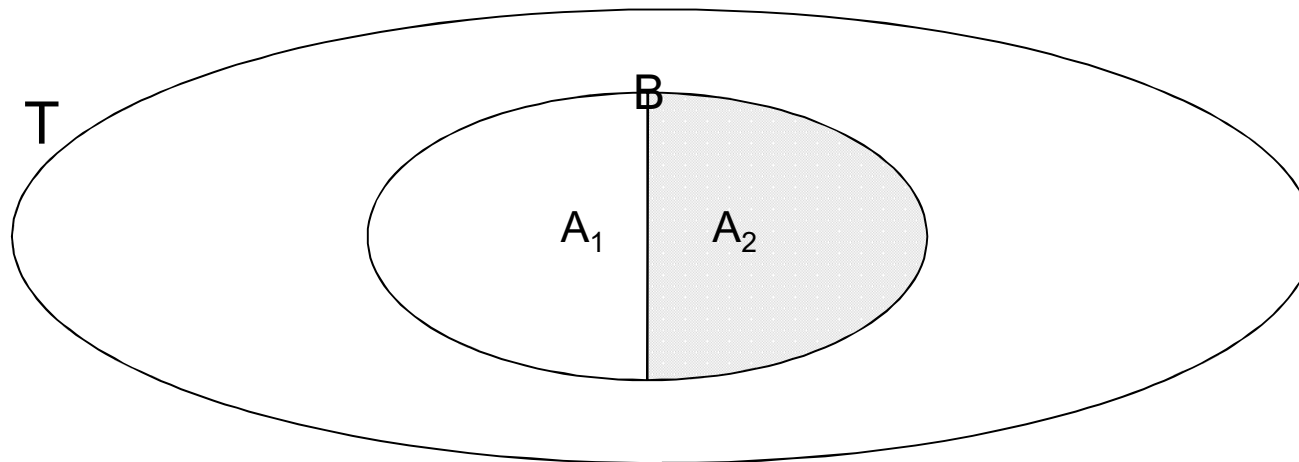


Modelowanie

Narzędzie 3: dziedziczenie

Dziedziczenie kompletne i rozłączne.

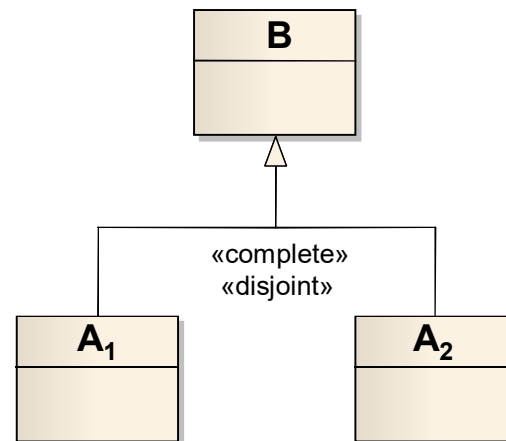
$$A_1 \sqcup A_2 \equiv B$$
$$A_1 \sqsubseteq \neg A_2$$



Modelowanie

Narzędzie 3: dziedziczenie

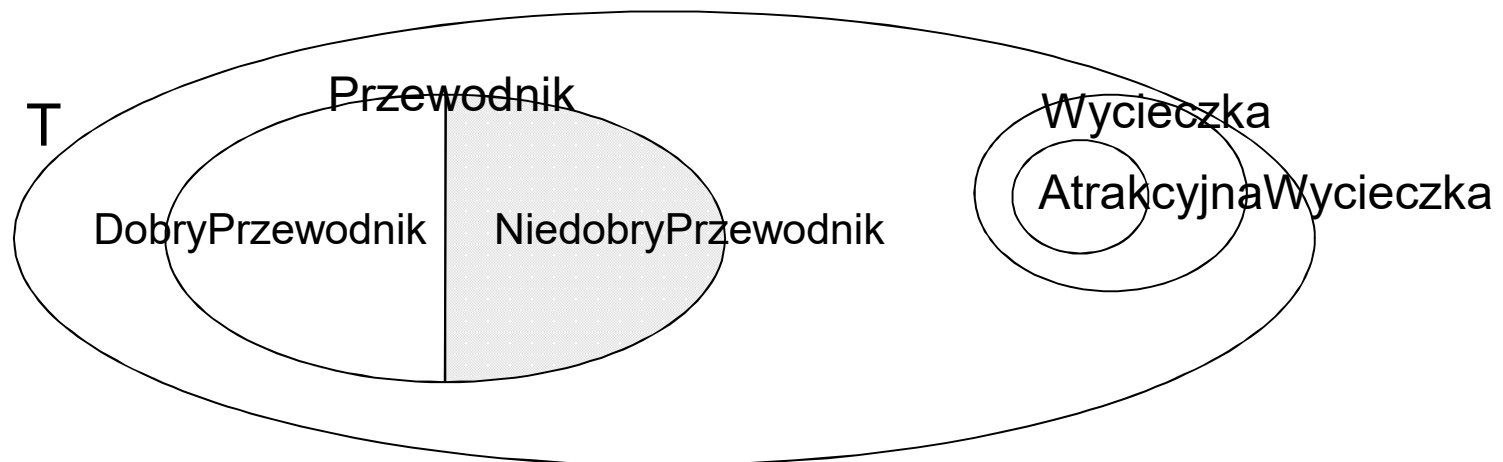
W UML-u jest możliwe zadeklarowanie kompletności i rozłączności dziedziczenia.



Modelowanie

Narzędzie 3: dziedziczenie

Ale uwaga! Tu łatwo popełnić błąd. Szczególnie w Protege'u.
Powiedzmy, że chcemy osiągnąć taki układ konceptów.



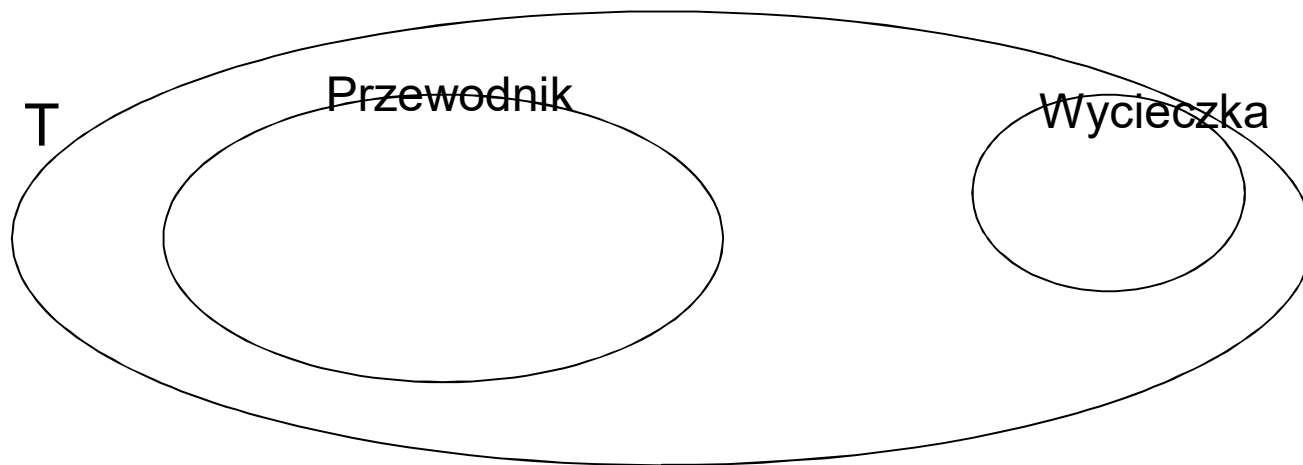
Modelowanie

Narzędzie 3: dziedziczenie

Formułujemy aksjomaty:

(1) Przewodnik \sqsubseteq \neg Wycieczka

(2) Przewodnik \equiv \exists proadzi.Wycieczka

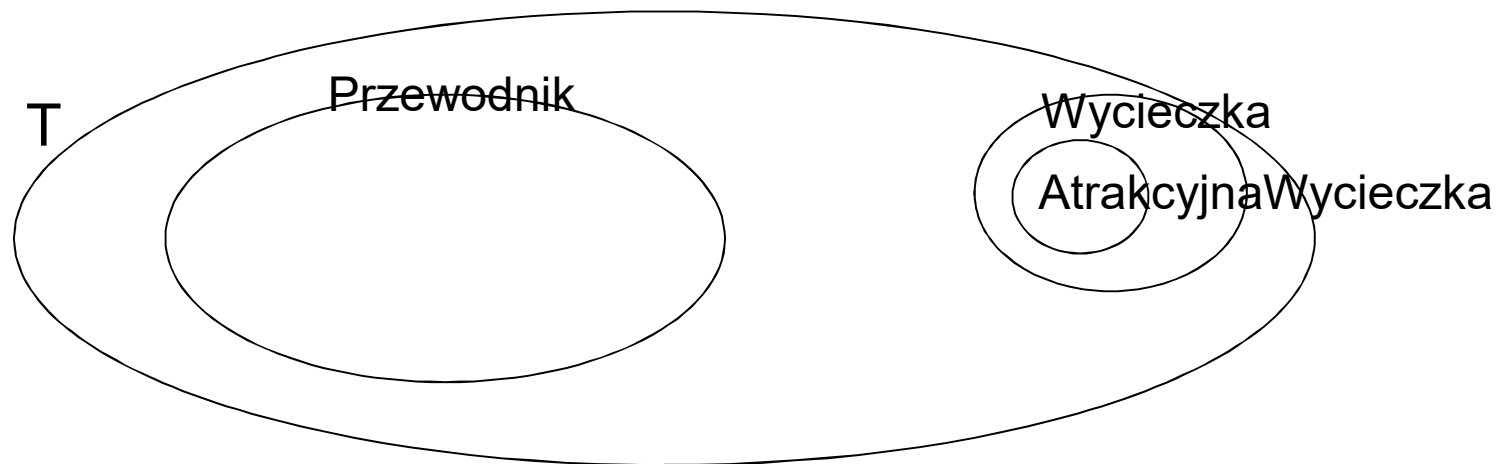


Modelowanie

Narzędzie 3: dziedziczenie

Formułujemy aksjomaty (w syntaktyce manchesterskiej):

- (1) Przewodnik \sqsubseteq \neg Wycieczka
- (2) Przewodnik \equiv \exists prowadzi.Wycieczka
- (3) AtrakcyjnaWycieczka \sqsubseteq Wycieczka

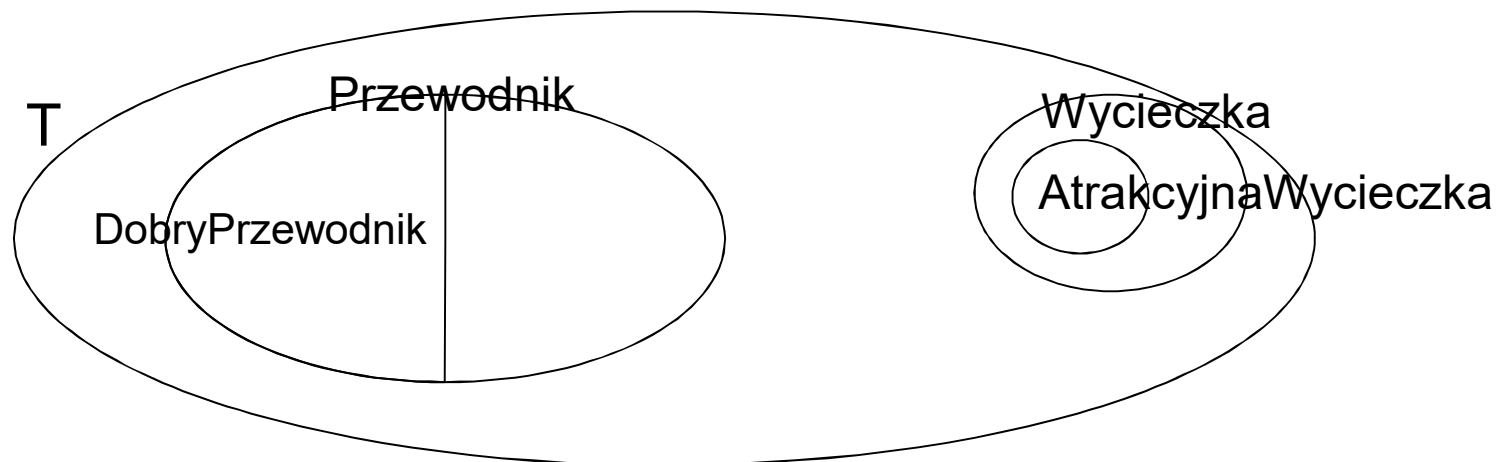


Modelowanie

Narzędzie 3: dziedziczenie

Formułujemy aksjomaty (w syntaktyce manchesterskiej):

- (1) Przewodnik \sqsubseteq \neg Wycieczka
- (2) Przewodnik \equiv \exists prowadzi.Wycieczka
- (3) AtrakcyjnaWycieczka \sqsubseteq Wycieczka
- (4) DobryPrzewodnik \equiv \exists prowadzi.AtrakcyjnaWycieczka



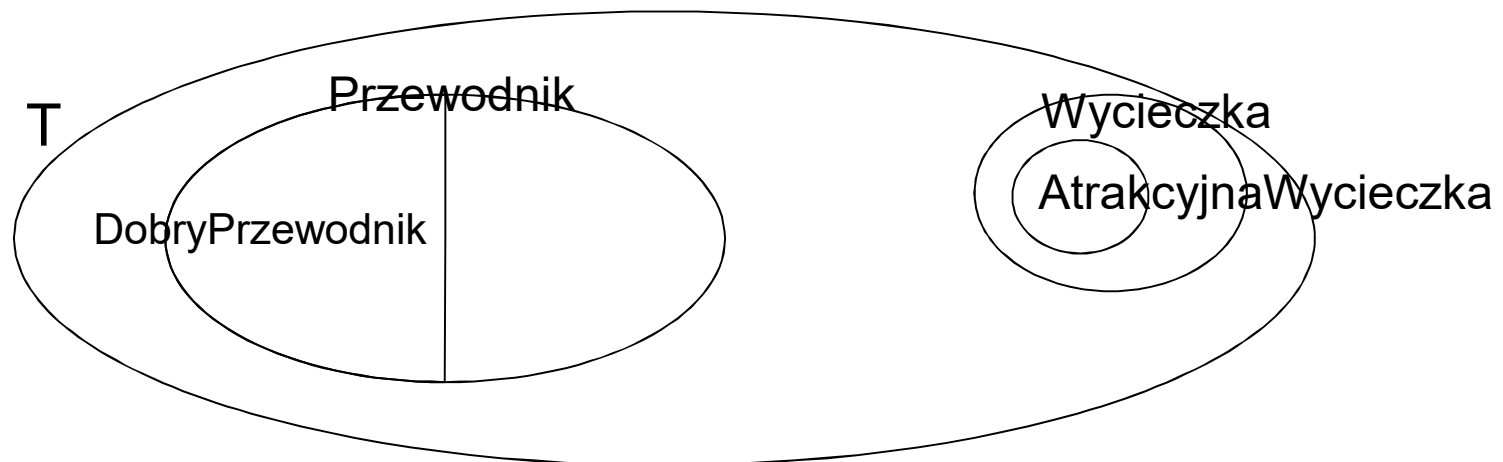
Modelowanie

Narzędzie 3: dziedziczenie

Formułujemy aksjomaty (w syntaktyce manchesterskiej):

- (1) Przewodnik \sqsubseteq \neg Wycieczka
- (2) Przewodnik \equiv \exists prowadzi.Wycieczka
- (3) AtrakcyjnaWycieczka \sqsubseteq Wycieczka
- (4) DobryPrzewodnik \equiv \exists prowadzi.AtrakcyjnaWycieczka
- (5) NiedobryPrzewodnik \equiv $\neg\exists$ prowadzi.AtrakcyjnaWycieczka

Co się stanie po tym ostatnim aksjomacie?

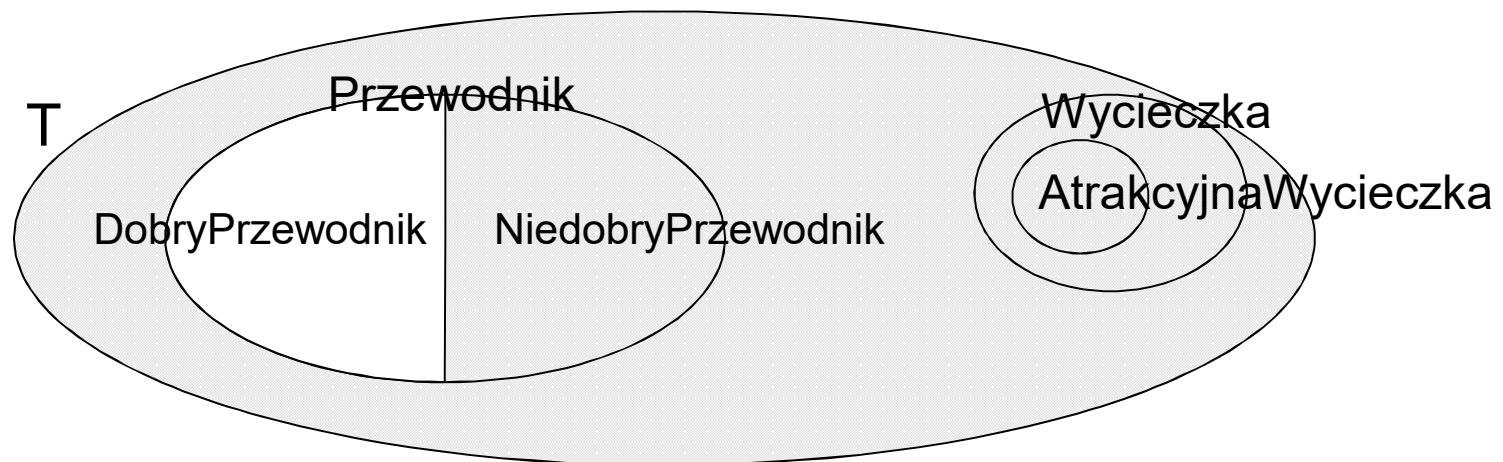


Modelowanie

Narzędzie 3: dziedziczenie

- (4) $\text{DobryPrzewodnik} \equiv \exists \text{prowadzi.AtrakcyjnaWycieczka}$
(5) $\text{NiedobryPrzewodnik} \equiv \neg \exists \text{prowadzi.AtrakcyjnaWycieczka}$

Koncept $\text{NiedobryPrzewodnik}$ pochłonie całą dziedzinę oprócz konceptu DobryPrzewodnik (łącznie z wycieczkami), ponieważ $A \sqcup \neg A \equiv T$.



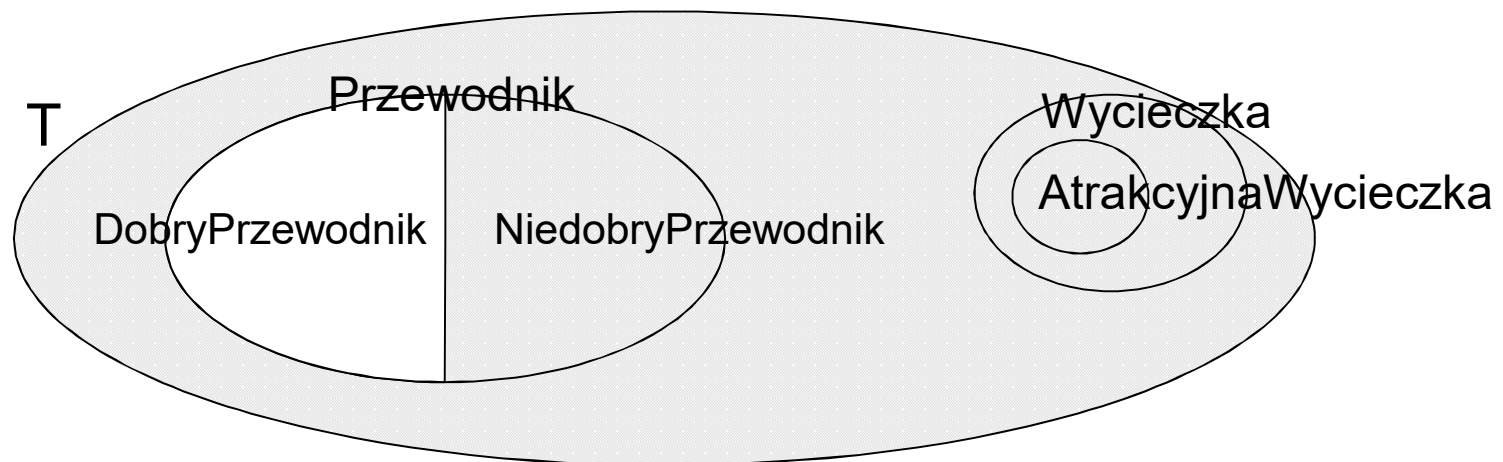
Modelowanie

Narzędzie 3: dziedziczenie

W Protege'u byłoby jeszcze gorzej:

ponieważ koncepty DobryPrzewodnik i NiedobryPrzewodnik zapewne zdefiniowalibyśmy jako dziedziczące od konceptu Przewodnik, to cała dziedzina stałaby się niespełnialna. Prawdziwe bowiem stałoby się zdanie: Przewodnik \equiv T.

To oraz aksjomat (1) spowodowałyby, że koncept Wycieczka stałby się niespełnialny (znalazłby się poza dziedziną). Na podstawie (2) niespełnialny stałby się koncept Przewodnik (gdyż nie mogą istnieć osobniki powiązane rolą z nieistniejącymi osobnikami), czyli Top.



Modelowanie

Narzędzie 3: dziedziczenie

Metoda ***OntoClean*** została stworzona w roku 2001 przez N. Guarino i C. A. Welty'ego.

Metoda ta służy do walidacji taksonomii.

(*Taksonomia*, to inaczej *drzewo dziedziczenia*).

Taksonomie mogą być tworzone nie tylko przez klasy, ale też przez relacje wyższych stopni (np. dziedziczenie ról). Dlatego przy omawianiu metody OntoClean zamiast słowa *klasa* będziemy używać słowa *właściwość*.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean

Metawłaściwości (właściwości właściwości):

- Trwałość (ang. *rigidity*, *essentiality*),
- Tożsamość (ang. *identity*)
- Całość (ang. *unity*)
- Zależność (ang. *dependency*)

Metawłaściwości pozwalają sformułować reguły dla procesu tworzenia taksonomii, których zadaniem jest odpowiednie pokierowanie tym procesem. Jednocześnie reguły te wyznaczają kryteria oceny jakości taksonomii.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Trwałość

Właściwość jest **trwała**, jeśli dla każdej encji, która ją wykazuje, jest ona *koniecznie* (czyli we wszystkich możliwych interpretacjach) jej właściwością.

Odnosząc to do danej ontologii, właściwość jest trwała, jeśli twórcy tej ontologii wykluczają jakąkolwiek możliwość takiego jej rozwoju, który pewnym encjom odebrałby tę właściwość.

Taką właściwością jest na przykład *bycie człowiekiem*. Jeśli dana encja wykazuje taką właściwość (jeśli dany obiekt jest człowiekiem), to będzie ją wykazywała zawsze (to musi nim być zawsze).

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Trwałość

Właściwość jest ***niepewnie trwała***, jeżeli wśród wszystkich encji wykazujących tę właściwość może zaistnieć taka, dla której nie jest ona trwała. Po prostu gdy istnieje jakakolwiek wątpliwość, że właściwość jest trwała, to jest ona niepewnie trwała.

Przykładem może być tutaj właściwość *bycia arcydziełem literackim*. Najczęściej arcydzieło literackie pozostaje arcydziełem na zawsze, a upływający czas jedynie potwierdza jego znaczenie. Może się jednak zdarzyć, że jakieś dzieło literackie, mimo że początkowo wysoko cenione, przestaje być takim po upływie jakiegoś czasu.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Trwałość

Właściwość jest ***nietrwała***, jeśli dla każdej encji wykazującej tę właściwość nie jest ona trwała. W przypadku tej właściwości istnieje pewność, że każda encja ją wykazująca w jakichś warunkach może ją stracić.

Przykładem może być właściwość *bycia studentem*. Wiadomo, że każdy student jest nim jedynie w pewnym okresie czasu.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Tożsamość

Mówiąc o metawłaściwości tożsamości, musimy najpierw zdefiniować, czym jest **kryterium tożsamości**.

Kryterium tożsamości, to relacja binarna, nazwijmy ją ρ , która stwierdza równość dwóch encji:

$$\rho(x, y) \leftrightarrow x = y$$

Inaczej mówiąc, jeśli para (pozornie) dwóch encji należy do relacji ρ (spełnia kryterium tożsamości), to jest to ta sama encja!

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Tożsamość

Kryterium tożsamości jest tworem związanym z właściwościami encji oraz zależnym od przyjętej dziedziny. Jeśli na przykład właściwością encji jest przynależność do klasy Człowiek, to relacja ρ o znaczeniu (dla szerszej dziedziny): [*mają taki sam odcisk palca*] lub (dla węższej dziedziny): [*mają taki sam numer PESEL*] może być zastosowana jako kryterium tożsamości.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Tożsamość

Właściwość może posiadać metawłaściwość:

- *przenoszenia kryterium tożsamości,*
- *nieprzenoszenia kryterium tożsamości,*
- *dostarczania kryterium tożsamości,*
- *niedostarczania kryterium tożsamości.*

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Tożsamość

Właściwość **przenosząca kryterium tożsamości**, dziedzicząc od właściwości dostarczającej tego kryterium, potrafi je przenieść. Może także dostarczyć własne kryterium tożsamości, ale to kryterium musi być kompatybilne do odziedziczonego.

Na przykład klasa Człowiek dostarcza kryterium tożsamości (np. odcisk palca), a klasa Student przenosi to kryterium (dziwne, ale studentów także można identyfikować wg odcisków palców). Może ona jednak dostarczać także własne kryterium (np. numer indeksu), ale kryterium to musi być kompatybilne (tzn. numer indeksu nie może być sprzeczny z odciskami palców).

Klasy nieprzenoszące (ani niedostarczające) kryteriów tożsamości, są bardzo ogólne np. Rzecz, Lokalizacja itp.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Całość

Istnieją różne rodzaje całości dla obiektów posiadających położenie w czasie i przestrzeni:

- *Całość topologiczna* (ang. *topological unity*): bazująca na fizycznym połączeniu w całość rozumianą topologicznie, na przykład bryła węgla, kamień, owoc.
- *Całość morfologiczna* (ang. *morphological unity*): bazująca na kształcie (na przykład kulistym — piłka), ale niekoniecznie stanowiąca całość topologiczną (na przykład konstelacja gwiazd).
- *Całość funkcjonalna* (ang. *functional unity*): bazująca na przeznaczeniu obiektu, na przykład komplet sztućców, bikini.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Całość

Właściwość, ze względu na metawłaściwość całości, dzielą się na:

- ***przenoszące kryterium całości***,
- ***nieprzenoszące kryterium całości***,
- posiadające metawłaściwość ***przeciwcałości***.

Klasa Człowiek przenosi wiele kryteriów całości, w zależności od tego, czy jej wystąpienia będziemy traktować jako byty fizyczne (części ciała), czy jako byty psychiczne (cechy psychiki).

Klasa Podatnik symbolizuje właściwość, która nie przenosi jednolitego kryterium całości dla wszystkich encji, gdyż obejmuje zarówno ludzi, jak i przedsiębiorstwa.

Metawłaściwość przeciwcałości posiadają rzeczy niepoliczalne: Woda, Materia.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – Zależność

Właściwość φ jest **zależna** od właściwości ψ , jeżeli dla każdej encji x wykazującej właściwość φ *koniecznie* istnieje jakaś encja y wykazująca właściwość ψ , która nie jest ani częścią, ani nie konstytuuje x .

W przeciwnym przypadku właściwość jest **niezależna**.

Klasa Rodzic symbolizuje właściwość zewnętrznie zależną, gdyż istnieje właściwość *bycia dzieckiem*, od której zewnętrznie zależy *bycie rodzicem*.

W przypadku właściwości *bycia człowiekiem* właściwość *posiadania serca* nie jest właściwością zewnętrznie zależną, gdyż serce jest częścią ciała ludzkiego (konstytuuje ciało ludzkie).

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean

Reguły wynikające z metawłaściwości:

1. Właściwość **nietrwała** nie może być dziedziczona przez **trwałą**.
2. Właściwość **przenosząca kryterium tożsamości** nie może być dziedziczona przez **nieprzenoszącą**.
3. Właściwość **przenosząca kryterium całości** nie może być dziedziczona przez **nieprzenoszącą**.
4. Właściwość będąca **przeciwczołością** nie może być dziedziczona przez właściwość **przenoszącą kryterium całości**.
5. Właściwość **zależna** nie może być dziedziczona przez **niezależną**.
6. Właściwości o niekompatybilnym kryterium tożsamości lub całości muszą być rozłączne.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean

Inaczej mówiąc:

1. Wyżej w hierarchii dziedziczenia powinny znajdować się klasy „trwalsze”.
2. Wyżej w hierarchii dziedziczenia powinny być klasy nie przenoszące kryterium tożsamości, niżej dostarczające tego kryterium, jeszcze niżej – przenoszące.
3. Idąc w dół hierarchii dziedziczenia nie możemy utracić kryterium całości.
4. Kryterium całości nie może się pojawić wśród klas dziedziczących od klas posiadających metawłaściwość przeciwcałości.
5. Idąc w dół hierarchii dziedziczenia klasa zależna nie może stać się niezależna.
6. Właściwości o niekompatybilnym kryterium tożsamości lub całości muszą być rozłączne (to bez zmian).

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean

Zasady dodatkowe:

- I. Każdy element dziedziny musi wykazywać jakąś właściwość przenoszącą kryterium tożsamości
- II. Jeśli dwie encje są tożsame, to muszą wykazywać jakąś wspólną właściwość, która pozwala to stwierdzić.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean

W poprawnie zdefiniowanej taksonomii wydzielamy trzy rodzaje właściwości:

1. Kategorie

Właściwości nie przenoszące kryterium tożsamości. Ponieważ zgodnie z zasadą 2) nie mogą dziedziczyć od właściwości przenoszących kryterium tożsamości, stanowią najwyższy poziom taksonomii i są prymitywne w takim sensie, że nie narzucają żadnych warunków przynależności. Kategorie tworzą archetypy, takie jak Rzecz, Lokalizacja itp.

2. Typy

Najważniejsze właściwości, gdyż dostarczają kryterium tożsamości. Mogą dziedziczyć tylko od kategorii lub innych typów, które dostarczają kompatybilnego kryterium tożsamości. Każda encja musi należeć do jakiegoś typu. Przykładowe typy: Zwierzę, Woda itp.

3. Quasi-typy

Dostarczają tzw. kryterium przynależności. Kryterium tożsamości mówi, czym różnią się encje, kryterium przynależności mówi, co je do siebie upodabnia. Każdy Quasi-typ musi dziedziczyć od przynajmniej jednego typu. Przykłady: Student, Kot itp.

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – przykład

W pewnej taksonomii zaistniała potrzeba wprowadzenia dwóch typów obiektów: *odstęp czasu* i *odcinek czasu*.

Odstęp czasu to: *godzina, dwie godziny, kwadrans*.

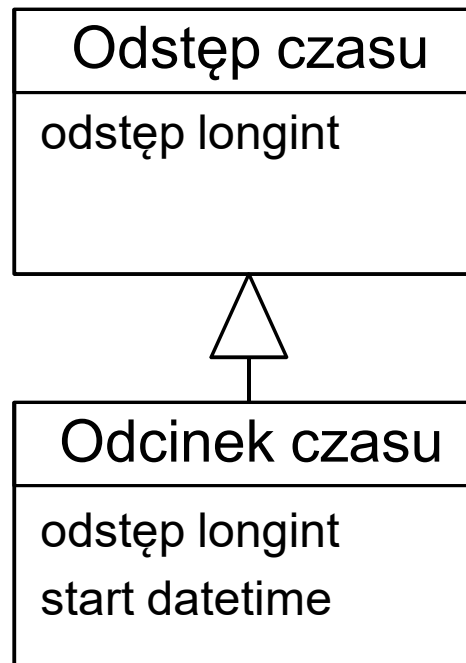
Odcinek czasu to: *godzina liczona od godz. 1:00 dnia 1.01.2012, kwadrans liczony od godz. 2:10 dnia 15.01.2012,*

Odcinek czasu, prócz czasu trwania, posiada jeszcze jeden atrybut: moment startu. **Czy więc powinien dziedziczyć od *odstępu czasu*?**

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – przykład



Model obiektowy: Dziedziczenie to rozszerzanie listy atrybutów?

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – przykład

Porównajmy kryteria tożsamości:

Odstęp czasu: dwie encje są tożsame, gdy mają ten sam czas trwania (np. godzinę). Nie mogą istnieć dwie różne encje o tym samym czasie trwania.

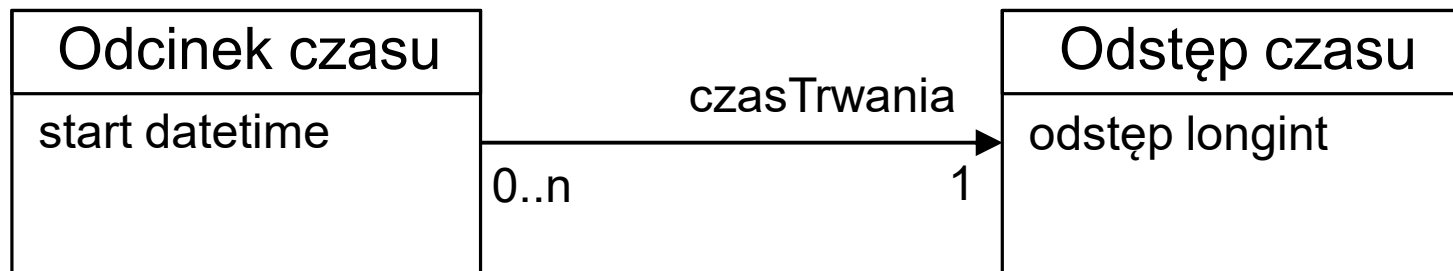
Odcinek czasu: dwie encje nie muszą być tożsame, gdy mają ten sam czas trwania (np. godzinę). Muszą jeszcze zaczynać się w tym samym momencie. Mogą więc istnieć dwie różne encje o tym samym czasie trwania.

Kryteria tożsamości obu typów są niekompatybilne. Zgodnie z regułą 6. takie typy muszą być rozłączne. *Odstęp czasu* może być jedynie komponentem *odcinka czasu* (połączenie rolą).

Modelowanie

Narzędzie 3: dziedziczenie

OntoClean – przykład



Model obiektowy: Klasy mogą jedynie być połączone asocjacją

Modelowanie

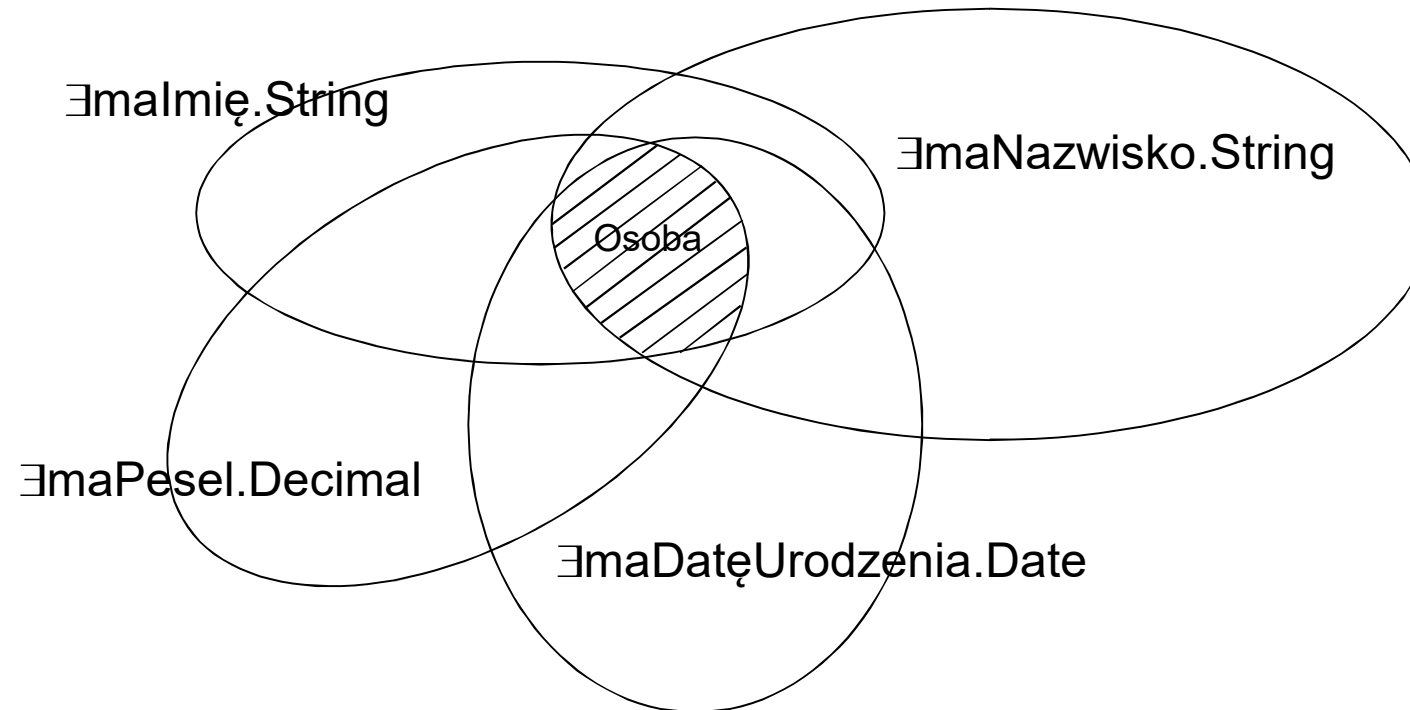
Narzędzie 4: atrybuty

Osoba
PESEL: Decimal
Imię: String
Nazwisko: String
DataUrodz: Data
...

W DL-u mamy kilka możliwości zdefiniowania atrybutów.

Modelowanie

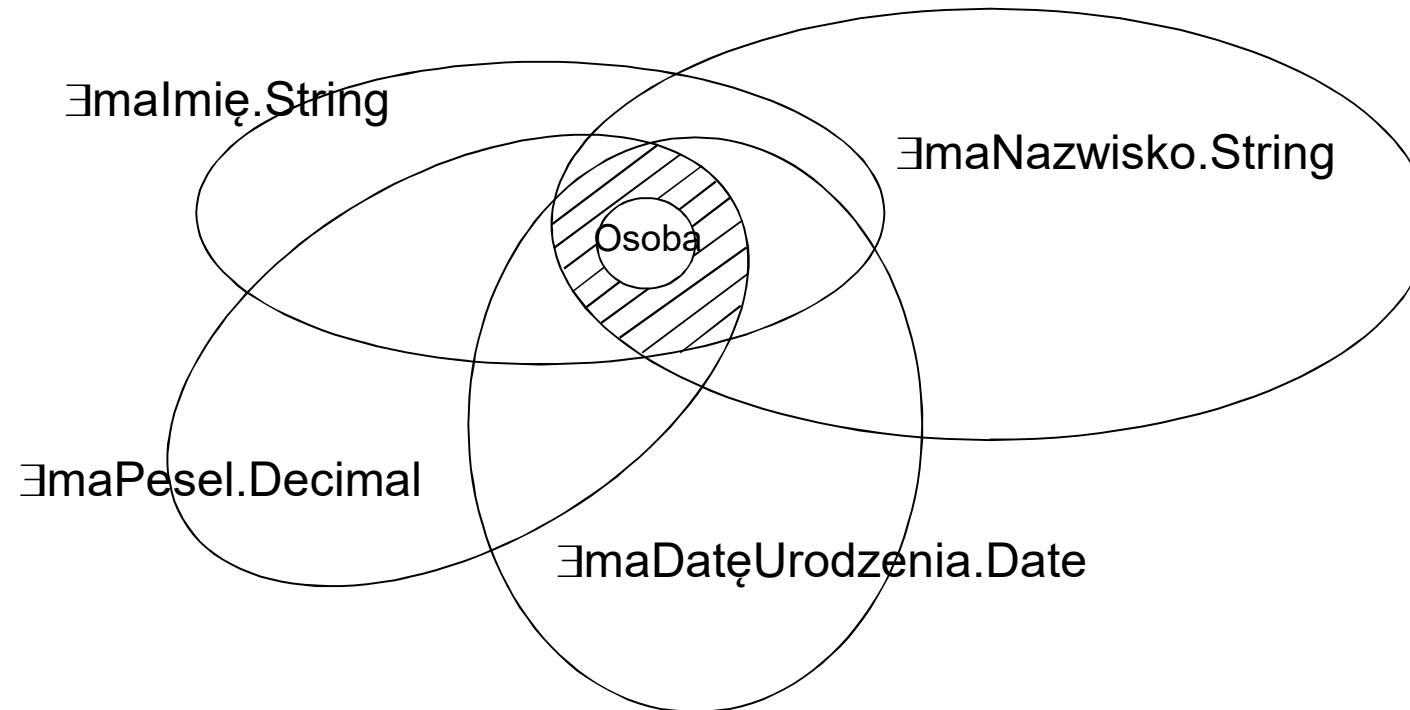
Narzędzie 4: atrybuty



$Osoba \equiv \exists Imię.String \cap \exists Nazwisko.String \cap$
 $\exists Pesel.Decimal \cap \exists DataUrodzenia.Date$

Modelowanie

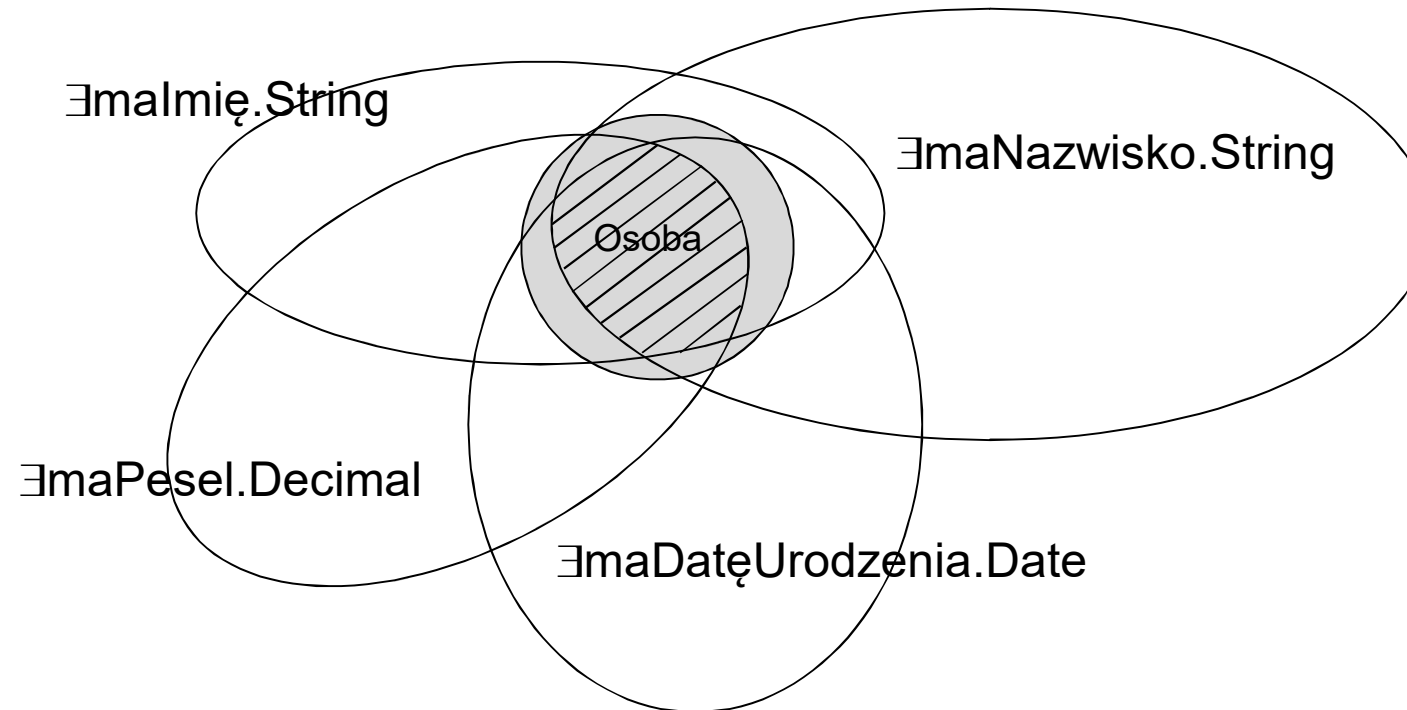
Narzędzie 4: atrybuty



$Osoba \sqsubseteq \exists Imię.String \sqcap \exists Nazwisko.String \sqcap$
 $\exists Pesel.Decimal \sqcap \exists DataUrodzenia.Date$

Modelowanie

Narzędzie 4: atrybuty



$Osoba \sqsubseteq \exists Imię.String \sqcap \exists Nazwisko.String \sqcap$
 $\exists Pesel.Decimal \sqcap \exists DataUrodzenia.Date$

Modelowanie

Narzędzie 4: atrybuty

Ale tu także należy uważać. Definicja taka:

$$\text{Osoba} \equiv \exists \text{maImię.String} \sqcap \exists \text{maNazwisko.String} \sqcap \\ \exists \text{maPesel.Decimal} \sqcap \exists \text{maDatęUrodzenia.Date}$$

wcale nie jest równoważna takiej:

$$\text{Osoba} \equiv \exists \text{maImię.String}$$
$$\text{Osoba} \equiv \exists \text{maNazwisko.String}$$
$$\text{Osoba} \equiv \exists \text{maPesel.Decimal}$$
$$\text{Osoba} \equiv \exists \text{maDatęUrodzenia.Date}$$

Modelowanie

Narzędzie 4: atrybuty

Ani taka:

Osoba \sqsupseteq \exists maImię.String \sqcap \exists maNazwisko.String \sqcap
 \exists maPesel.Decimal \sqcap \exists maDatęUrodzenia.Date

nie jest równoważna takiej:

Osoba \sqsupseteq \exists maImię.String
Osoba \sqsupseteq \exists maNazwisko.String
Osoba \sqsupseteq \exists maPesel.Decimal
Osoba \sqsupseteq \exists maDatęUrodzenia.Date

Modelowanie

Narzędzie 4: atrybuty

Ale taka:

Osoba \sqsubseteq \exists maImię.String \sqcap \exists maNazwisko.String \sqcap
 \exists maPesel.Decimal \sqcap \exists maDatęUrodzenia.Date

jest równoważna takiej:

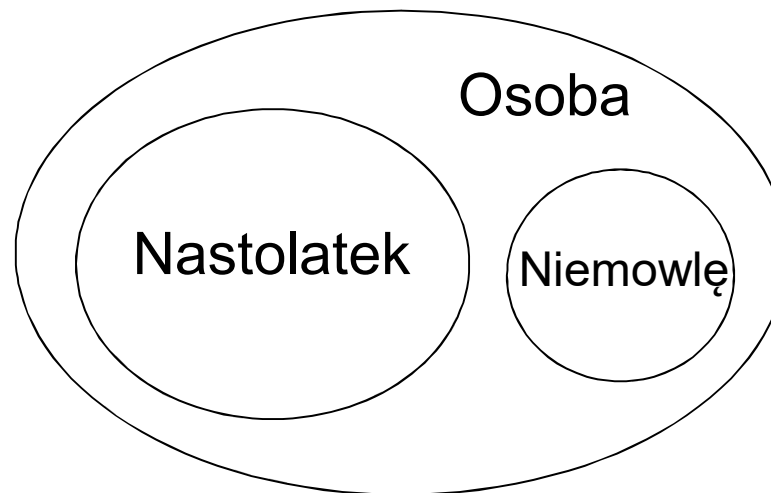
Osoba \sqsubseteq \exists maImię.String
Osoba \sqsubseteq \exists maNazwisko.String
Osoba \sqsubseteq \exists maPesel.Decimal
Osoba \sqsubseteq \exists maDatęUrodzenia.Date

Modelowanie

Narzędzie 5: kontekst

Na kształt tworzonego modelu zawsze silnie wpływa kontekst!

Na przykład poniższy model wydaje się dość oczywisty. A jednak jeśli model miałby być prawdziwy przez dłuższy okres czasu (np. 20 lat), to jak opisać w nim encję reprezentującą człowieka, który w tym czasie był i niemowlęciem i nastolatkiem?



Modelowanie

Narzędzie 5: kontekst

Najczęściej brane pod uwagę tzw. parametry kontekstowe:

1. Czas (moment i odcinek czasu)
2. Miejsce
3. Punkt widzenia (np. osoby, przedsiębiorstwa)
4. Świat możliwy (np. świat jakiejś opowieści, np. o Sherlocku Holmesie).

Tak naprawdę parametrów kontekstowych jest o wiele więcej.

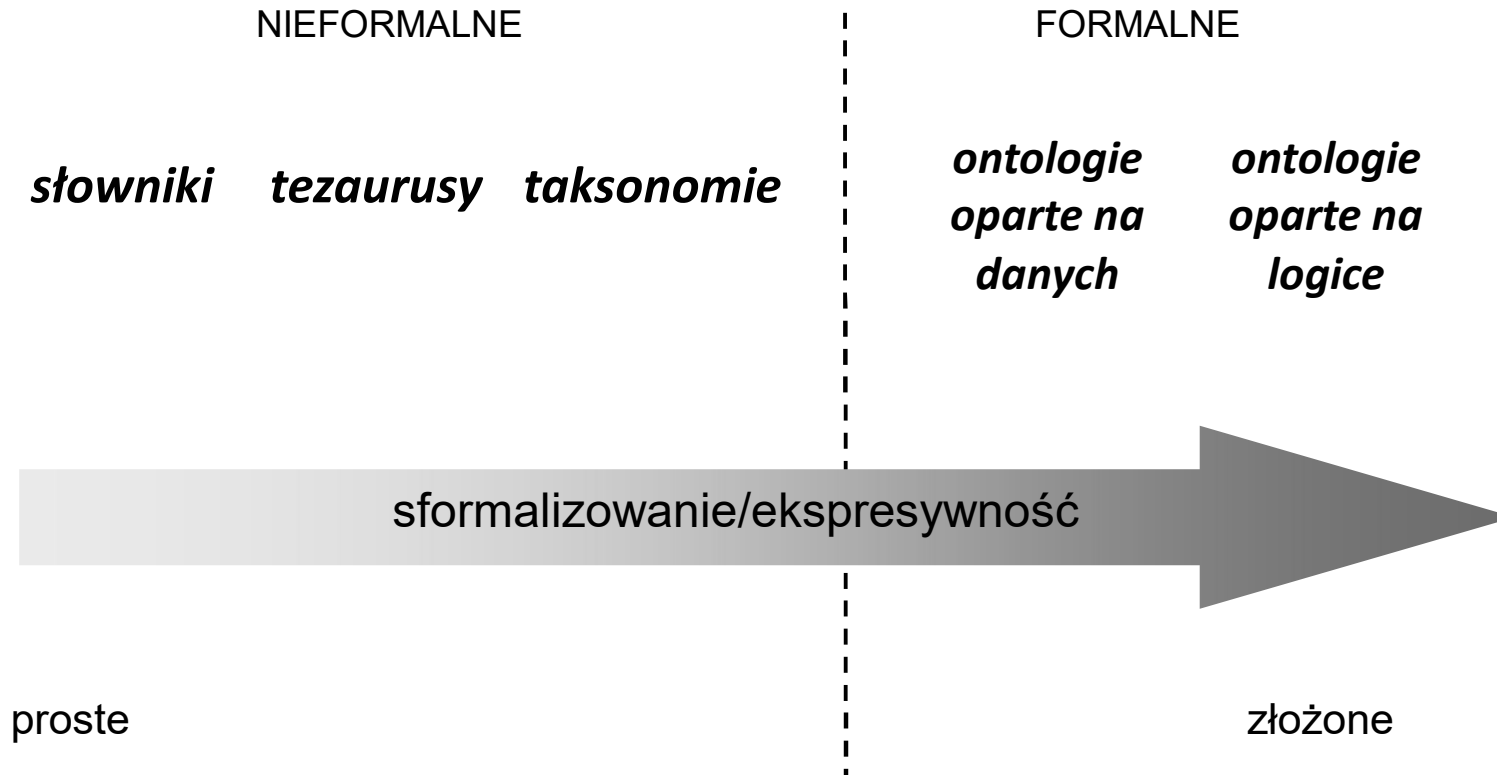
Stopień formalizacji

- Ontologie wysoce nieformalne:
 - wyrażone w języku naturalnym
- Ontologie semi-nieformalne:
 - ograniczenie języka naturalnego do ustalonego rodzaju wyrażeń,
 - możliwość użycia zmiennych;
- Ontologie semi-formalne:
 - użycie języka formalnego, ale bez możliwości wnioskowania, przeprowadzania dowodów (UML, XML, RDF itp.)
- Ontologie formalne:
 - języki logiki z możliwością wnioskowania (FOL, DL, Prolog)

Stopień złożoności

- Kontrolowane słowniki (listy uzgodnionych nazw)
- Glosariusze (słowniki z wyjaśnieniami w jęz. naturalnym)
- Systemy klasyfikacji (normy, klasyfikacja roślin)
- Tezaurusy (glosariusze wraz z relacjami między terminami: synonimia, antonimia, hiperonimia-hiponimia, czyli dziedziczenie, metonimia, czyli zastępowanie: np. czytam Słowackiego itp.)
- Taksonomie (glosariusze wraz z relacją subsumcji, dziedziczeniem)
- Ontologie „ramkowe” (diagram klas)
- Ontologie umożliwiające definiowanie relacji
- Teorie aksjomatyczne

Stopień formalizacji/złożoności



Źródło: K. Goczyła „Ontologie w systemach informatycznych”

Zastosowania ontologii

- Systemy ekspertowe
 - głównie bazy regułowe, najczęściej w medycynie
- Systemy agentowe
 - opis wiedzy agentów
- Wspomaganie projektowania systemów informatycznych
 - uzgodnienie opisu dziedziny
- Integracja systemów informatycznych
 - utworzenie wspólnej platformy semantycznej dla różnych schematów danych
- Przetwarzanie języka naturalnego
 - definicja znaczenia używanych słów
- Standaryzacja terminologii dziedzinowej
 - uzgodnienie wiedzy ekspertów
- Semantic Web
 - semantyczne znacznikowanie treści zawartych w Internecie

Ontologie stosowane

1-sza definicja

Ontologia jest teorią aksjomatyczną:

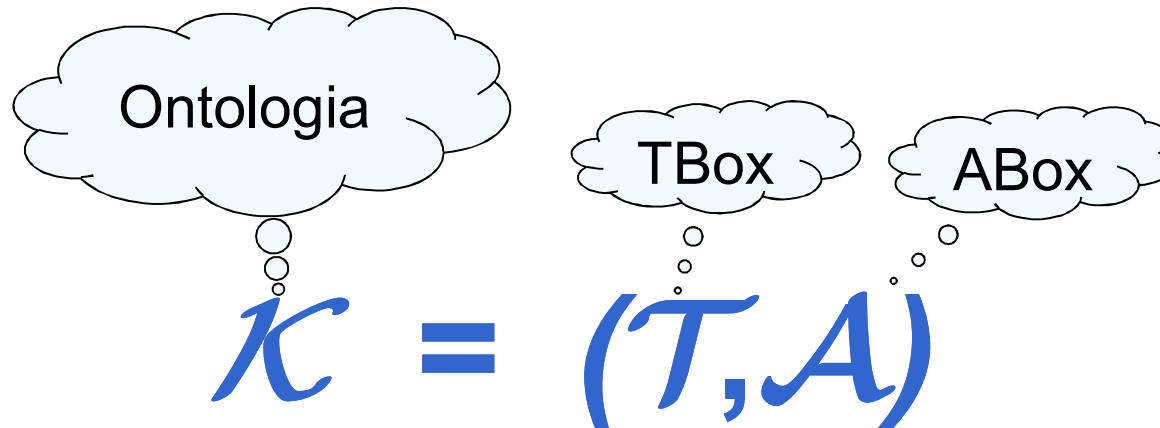
$$O = (L, A, R)$$

L – język

A – aksjomaty

R – reguły wnioskowania

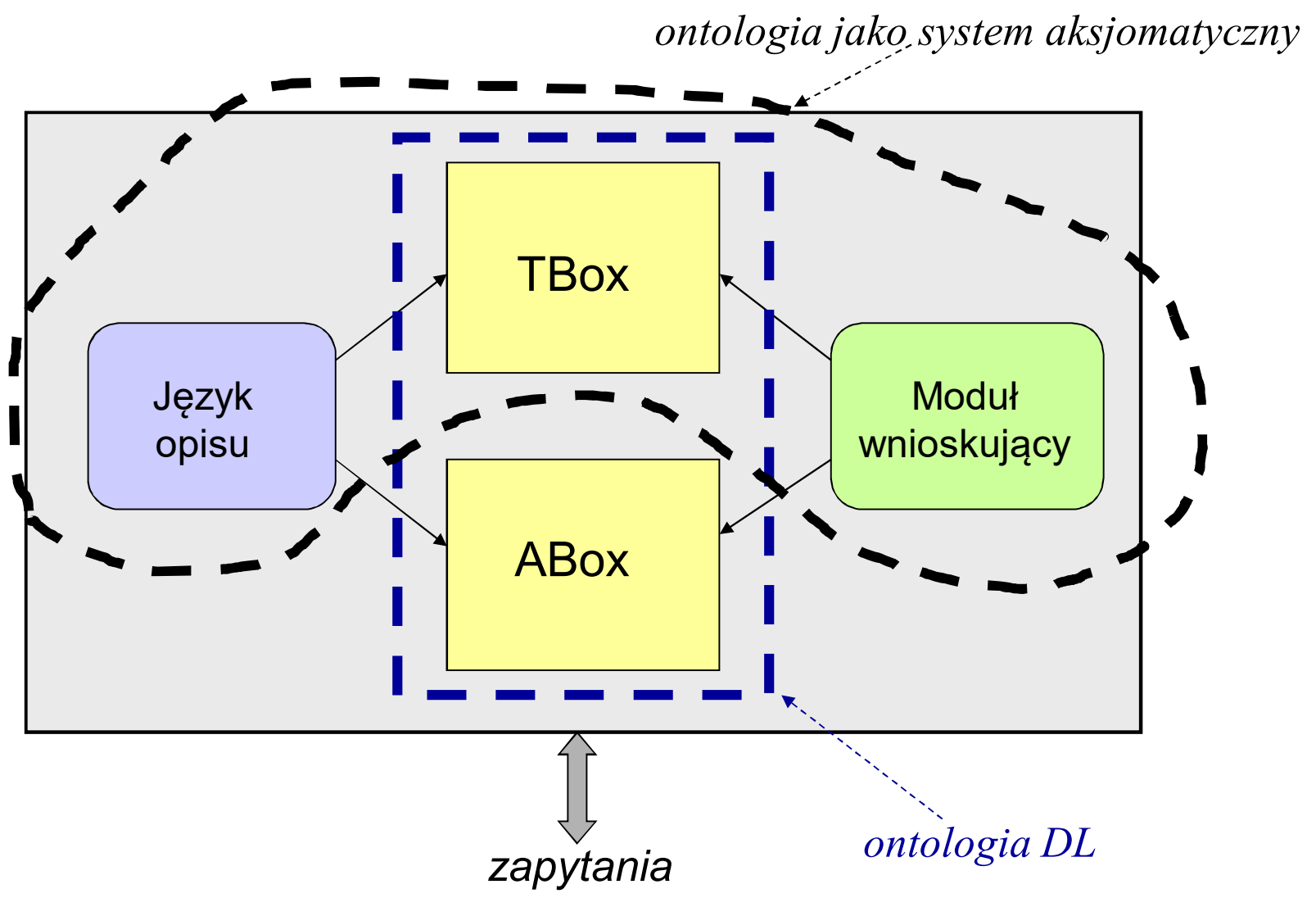
Ontologia DL - składniki



TBox (terminologia) – zawiera zdania terminologiczne (*aksjomaty*). Zdania te definiują *koncepty* i *role*.

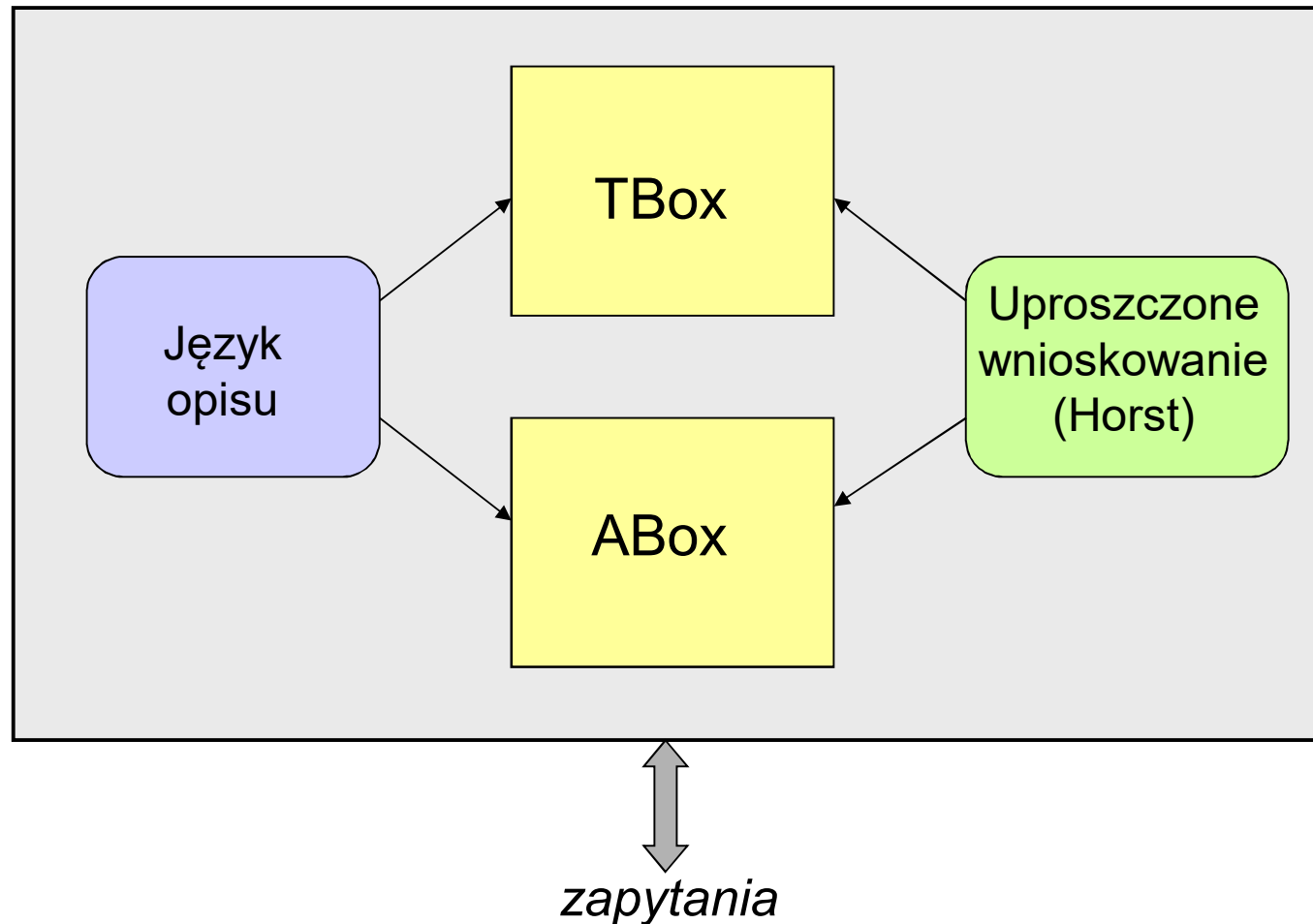
ABox (opis świata) – zawiera zdania opisujące fakty (*asercje*). Zdania te przypisują koncepty *osobnikom* i role *parom osobników*.

Baza wiedzy



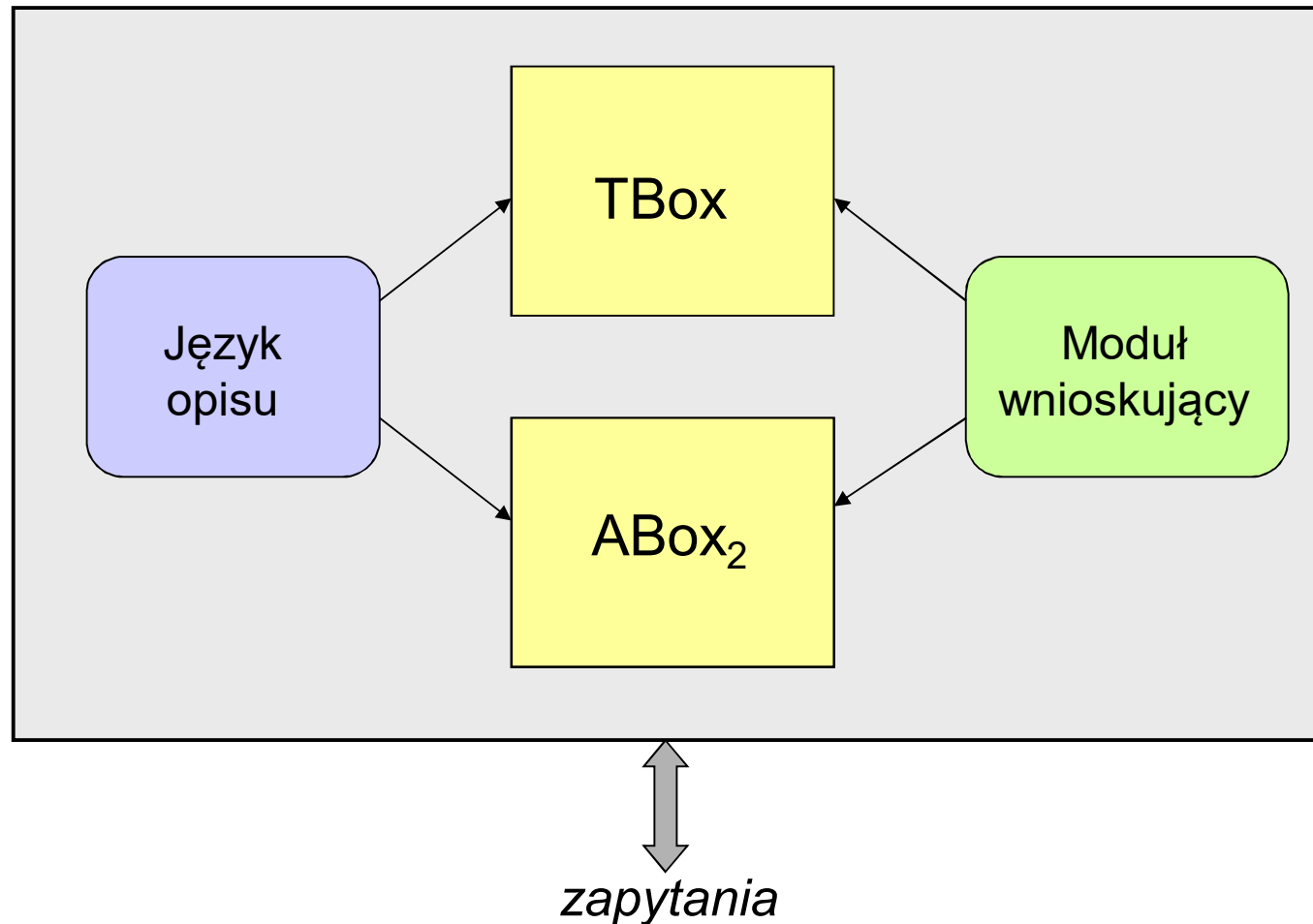
Scenariusze użycia

Składowanie danych: Abox stały



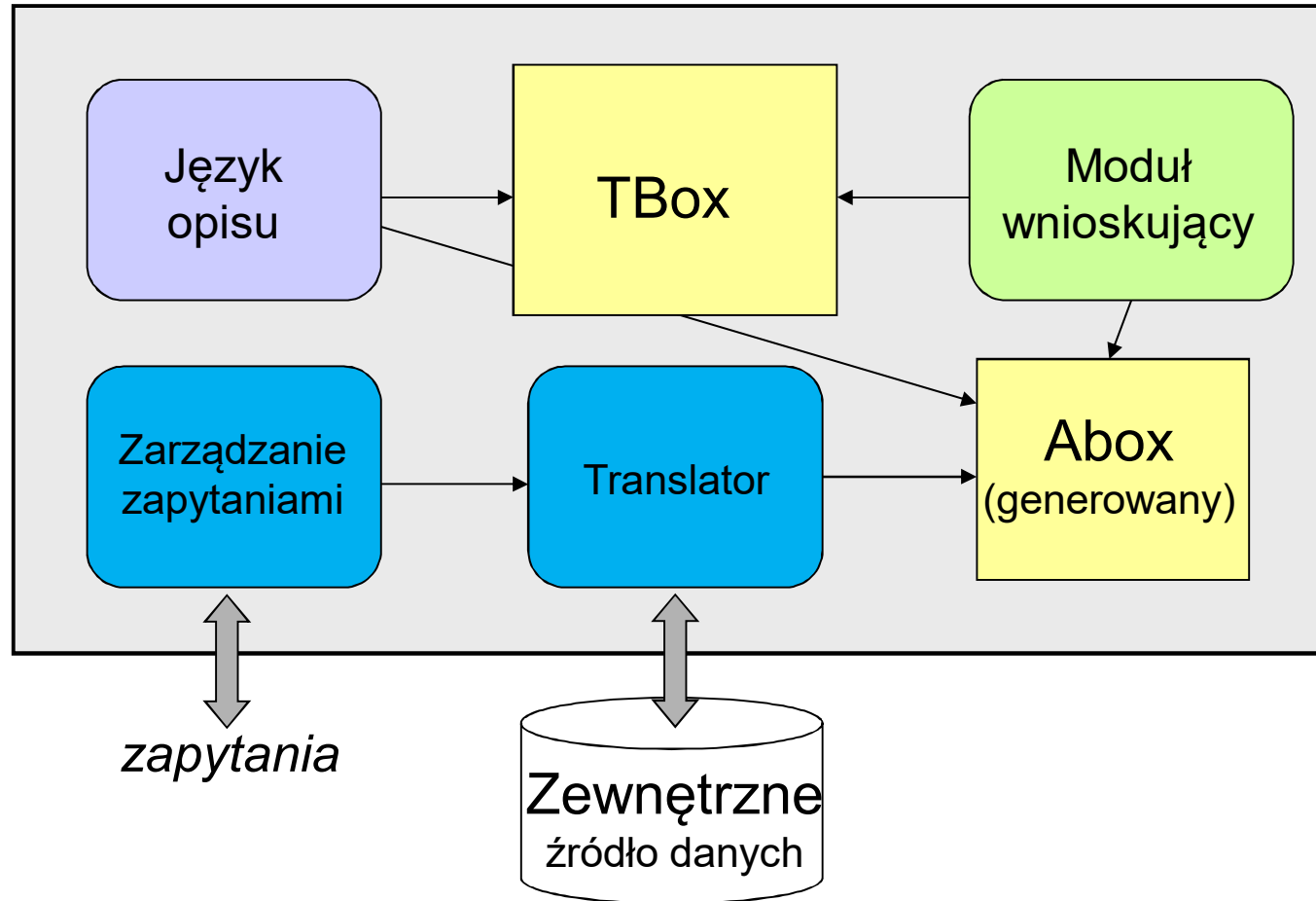
Scenariusze użycia

Systemy ekspertowe: Abox jako wsad



Scenariusze użycia

Systemy hybrydowe: pozyskiwanie faktów
z zewnętrznych źródeł danych



Poziom ogólności

- Ontologie aplikacji
 - ściśle związane z danym zastosowaniem
- Ontologie dziedzinowe
 - związane z daną dziedziną; wykraczają ponad jedno zastosowanie
- Ontologie ogólne
 - korzystają z doświadczeń ontologii filozoficznej; dla grupy dziedzin lub wszystkich dziedzin (fundacyjna)
- Ontologie reprezentacji
 - metaontologie (ontologie ontologii)

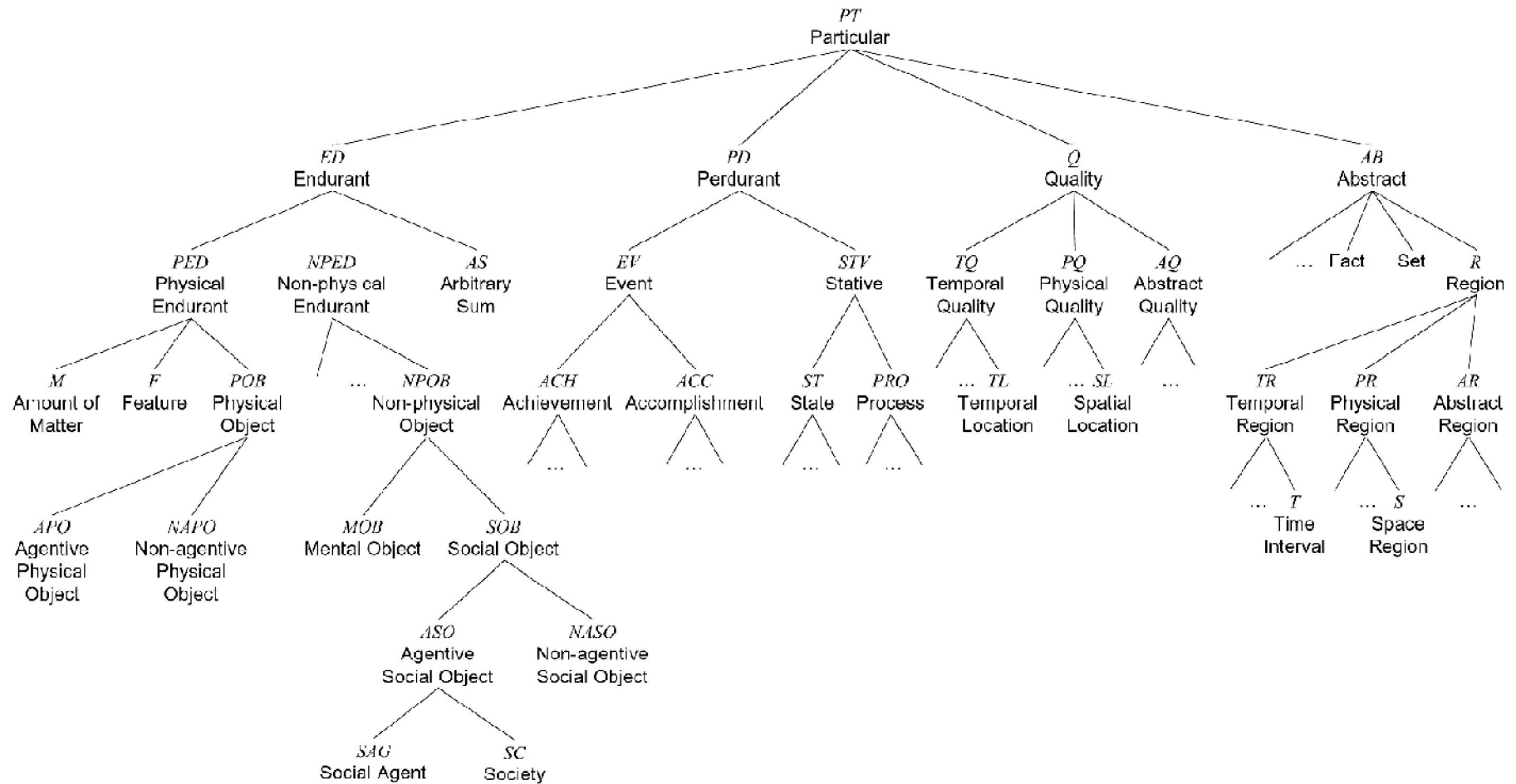
Powtórne użycie

Nazwa	Opis
DOLCE www.loa-cnr.it/DOLCE.html	Ontologia fundacyjna
SUMO http://www.ontologyportal.org	Ontologia fundacyjna
OpenCyc www.cyc.com/SUO/opencyc-ontology.txt	Ontologia wysokiego poziomu do analizy języka naturalnego
Dublin Core www.dublincore.org	Ontologia wysokiego poziomu głównie do opisu zasobów bibliotek
CIDOC CRM http://www.codoc-crm.org	Ontologia wysokiego poziomu do opisu ob. dziedzictwa kulturowego
FOAF http://www.foaf-project.org	Ontologia opisująca stosunki między ludźmi
SIOC http://rdfs.org/sioc/spec/	Ontologia do opisu serwisów społecznościowych



DOLCE

Ontologia fundacyjna wykorzystująca doświadczenia ontologii filozoficznej

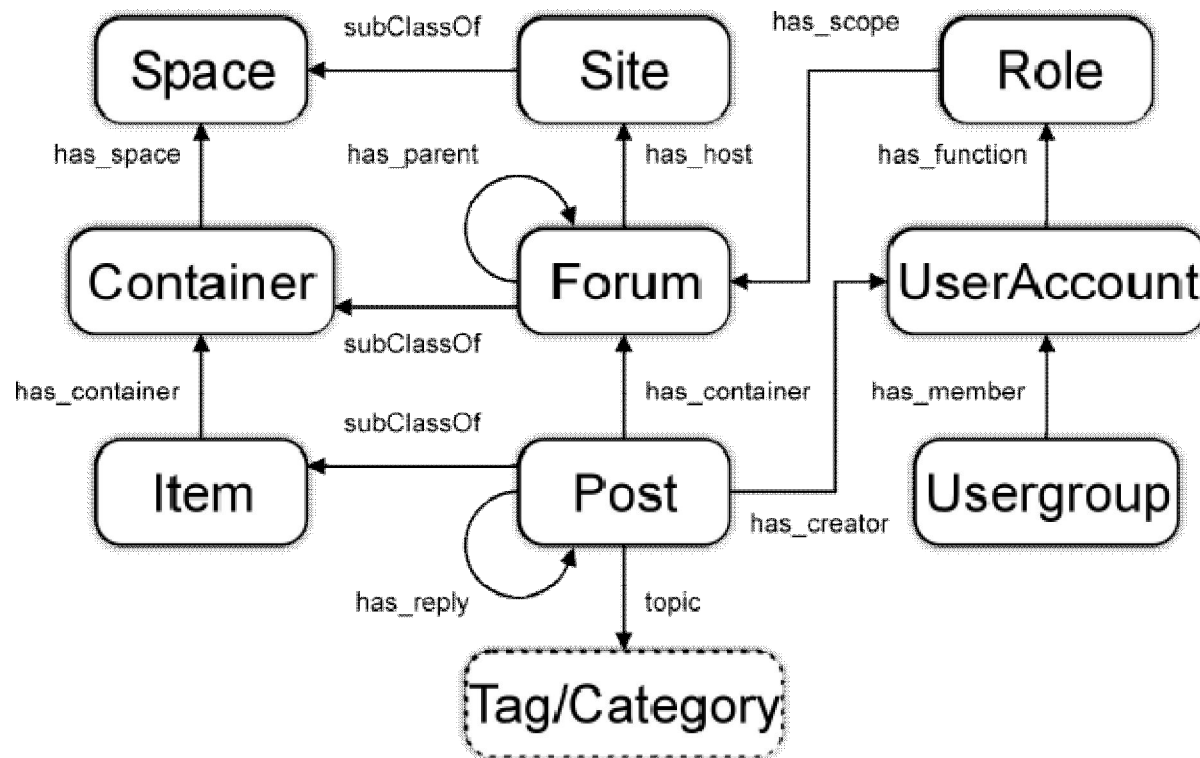


FOAF

- FOAF (Friend of a Friend) jest ontologią w zamyśle przeznaczona do opisywania stosunków między ludźmi (przede wszystkim na potrzeby tzw. *social networks*),
- FOAF jest popularna, gdyż wprowadza szereg terminów wykorzystywanych w wielu aplikacjach:
 - podmioty i osoby: foaf:Agent, foaf:Person,
 - właściwości osób i instytucji, takie jak adres e-mail (foaf:mbox) czy strona domowa (foaf:homepage), nazwisko (foaf:familyname)

SIOC

- SIOC jest ontologią przeznaczoną do opisu serwisów społecznościowych (or. *online community sites*),

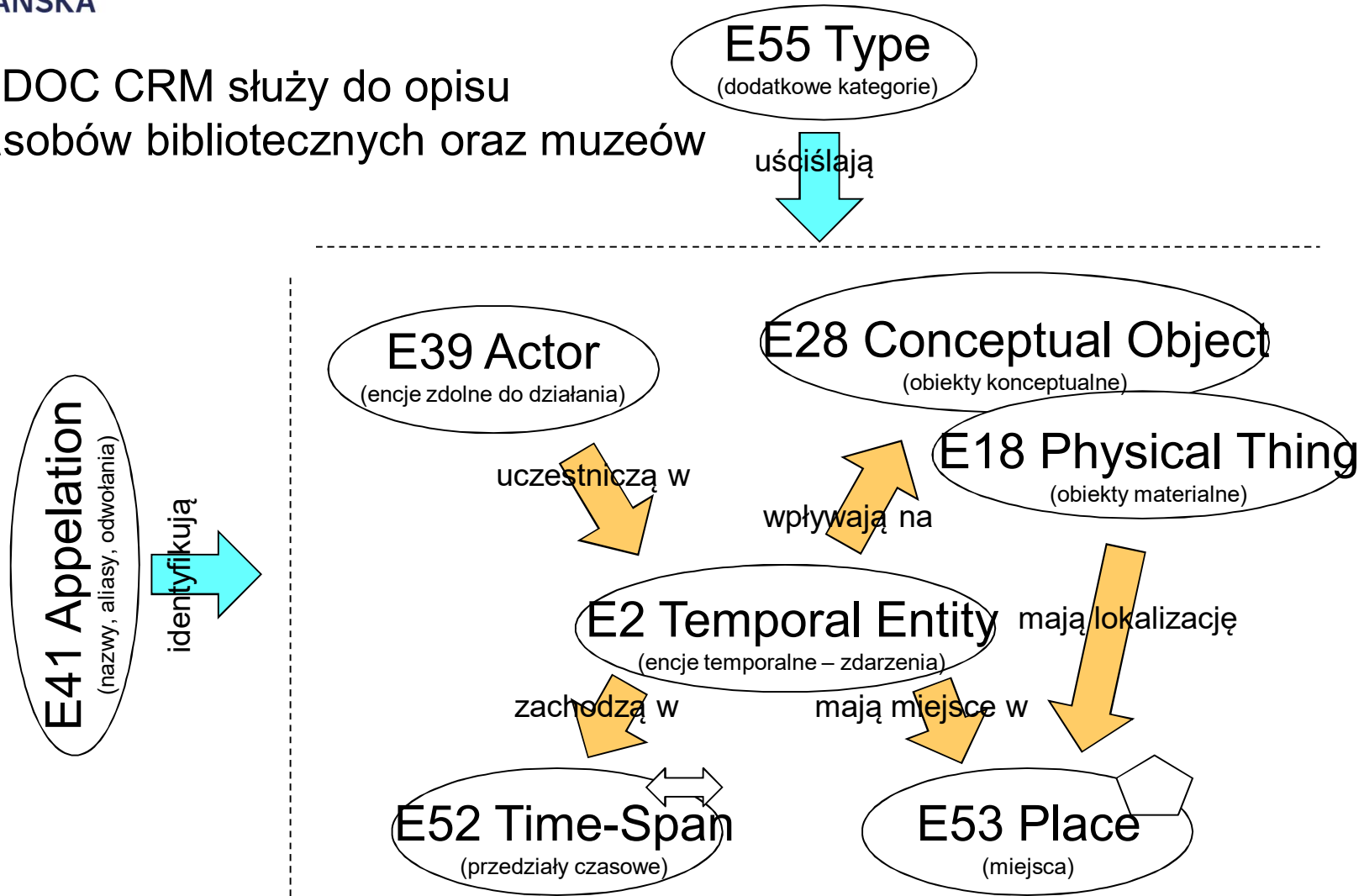


Dublin Core

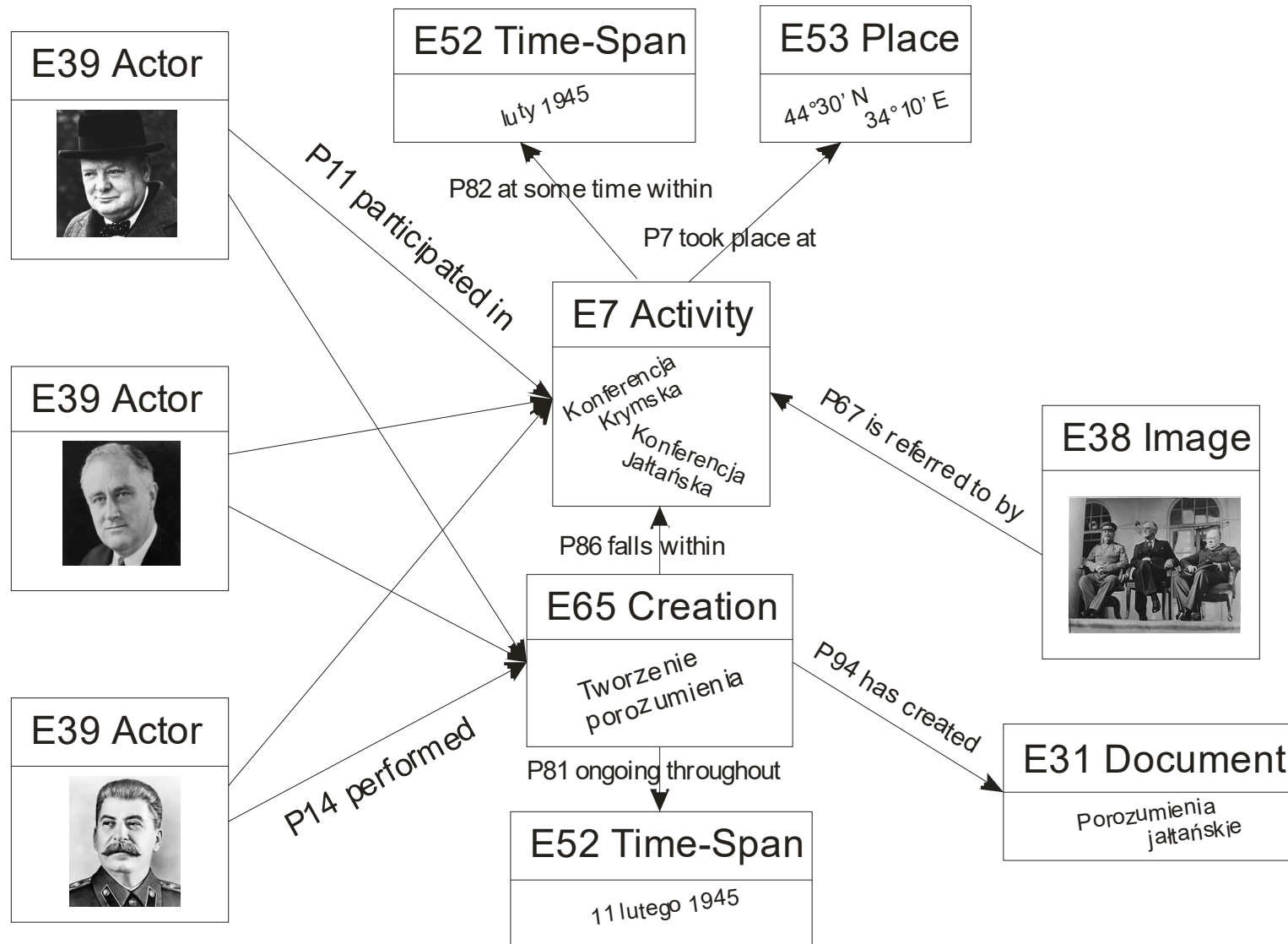
- Dublin Core to zestaw bardzo popularnych terminów do opisu metadanych, dostępny również w postaci ontologii, używany głównie do opisu zasobów bibliotek cyfrowych.
- Przykłady terminów Dublin Core:
 - temat (zasobu) dc:subject,
 - twórca (zasobu) dc:creator,
 - tytuł (zasobu) dc:title

CIDOC CRM

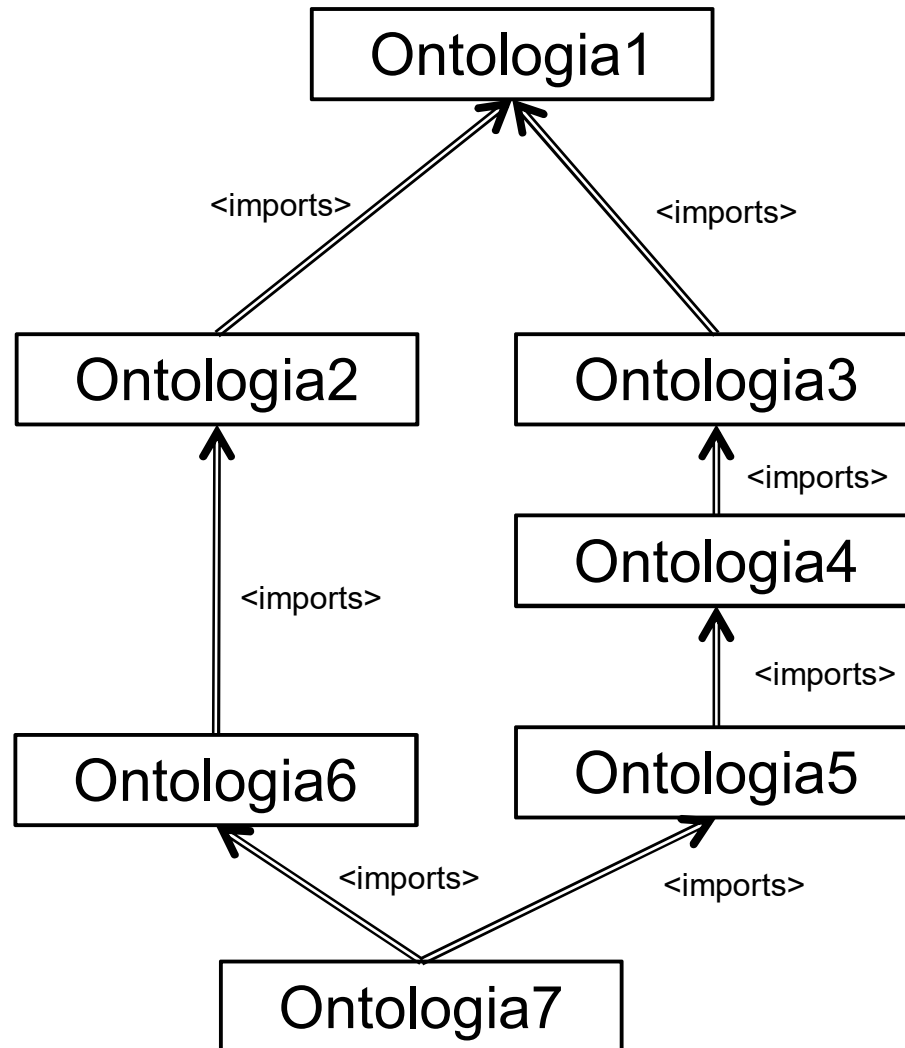
CIDOC CRM służy do opisu
zasobów bibliotecznych oraz muzeów



Przykład CRM



Powtórne użycie

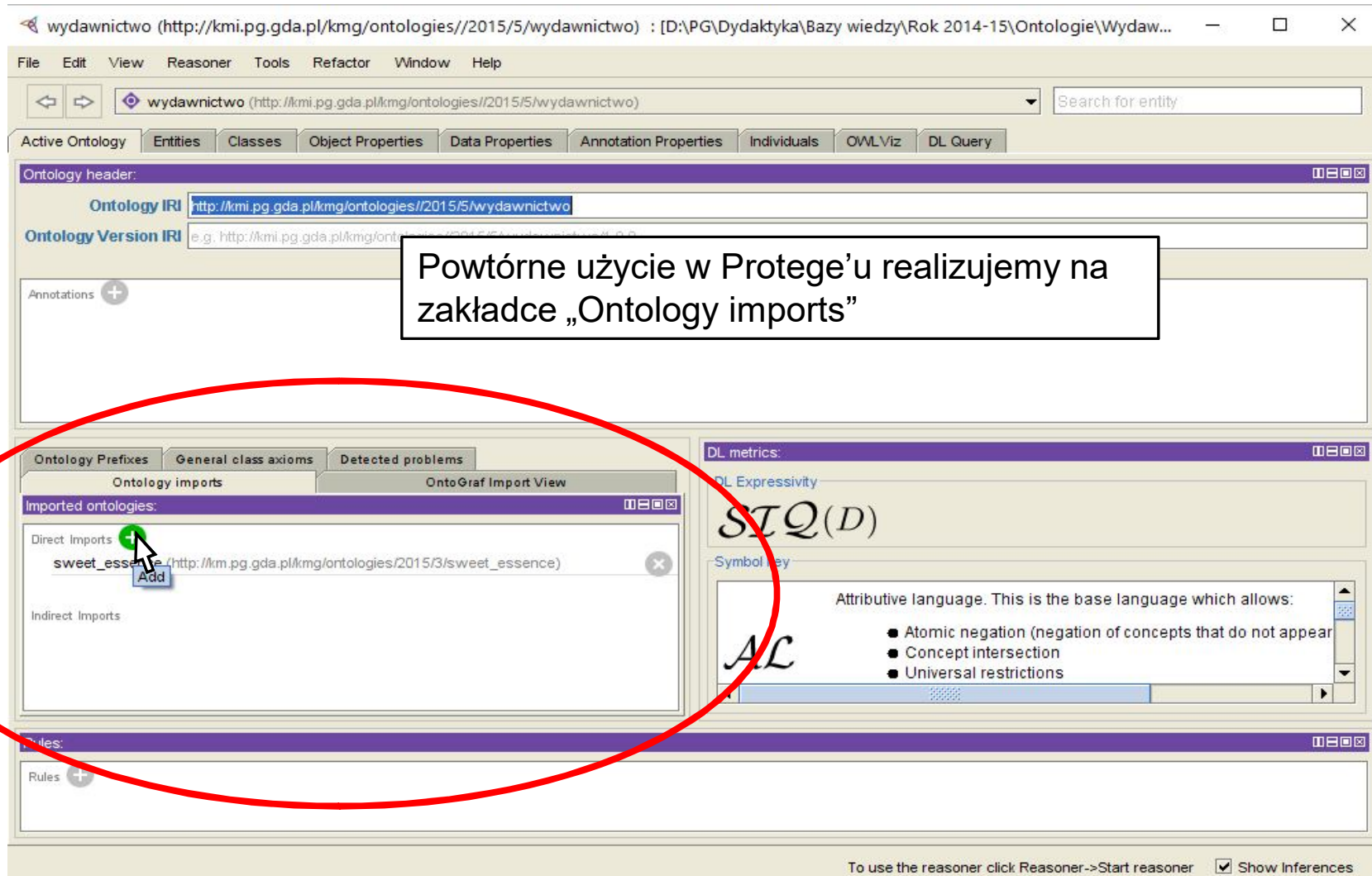


Powtórne użycie realizujemy najczęściej jako „import”, czyli wskazanie adresu IRI ontologii powtórnie użytej.

Język OWL zawiera odpowiednią strukturę służącą do tego celu.

W rezultacie importu otrzymujemy nową ontologię, która jest sumą ontologii importującej i importowanej

Powtórne użycie



The screenshot shows the Protege software interface. The main window title is "wydawnictwo (http://kmi.pg.gda.pl/kmg/ontologies//2015/5/wydawnictwo)". The "Ontology imports" tab is active, showing a list of imported ontologies. One ontology, "sweet_essence (http://kmi.pg.gda.pl/kmg/ontologies/2015/3/sweet_essence)", is listed under "Direct Imports". A red circle highlights this area. A text box with a black border and white background is overlaid on the interface, containing the text: "Powtórne użycie w Protege'u realizujemy na zakładce „Ontology imports”".

wydawnictwo (http://kmi.pg.gda.pl/kmg/ontologies//2015/5/wydawnictwo) : [D:\PG\Dydaktyka\Bazy wiedzy\Rok 2014-15\Ontologie\Wydaw...

File Edit View Reasoner Tools Refactor Window Help

wydawnictwo (http://kmi.pg.gda.pl/kmg/ontologies//2015/5/wydawnictwo) Search for entity

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWL Viz DL Query

Ontology header:

Ontology IRI <http://kmi.pg.gda.pl/kmg/ontologies//2015/5/wydawnictwo>

Ontology Version IRI e.g. <http://kmi.pg.gda.pl/kmg/ontologies//2015/5/wydawnictwo#20150501>

Annotations +

Ontology Prefixes General class axioms Detected problems

Ontology imports OntoGraf Import View

Imported ontologies:

Direct Imports

sweet_essence (http://kmi.pg.gda.pl/kmg/ontologies/2015/3/sweet_essence) Add

Indirect Imports

DL metrics:

DL Expressivity

$STQ(D)$

Symbolic

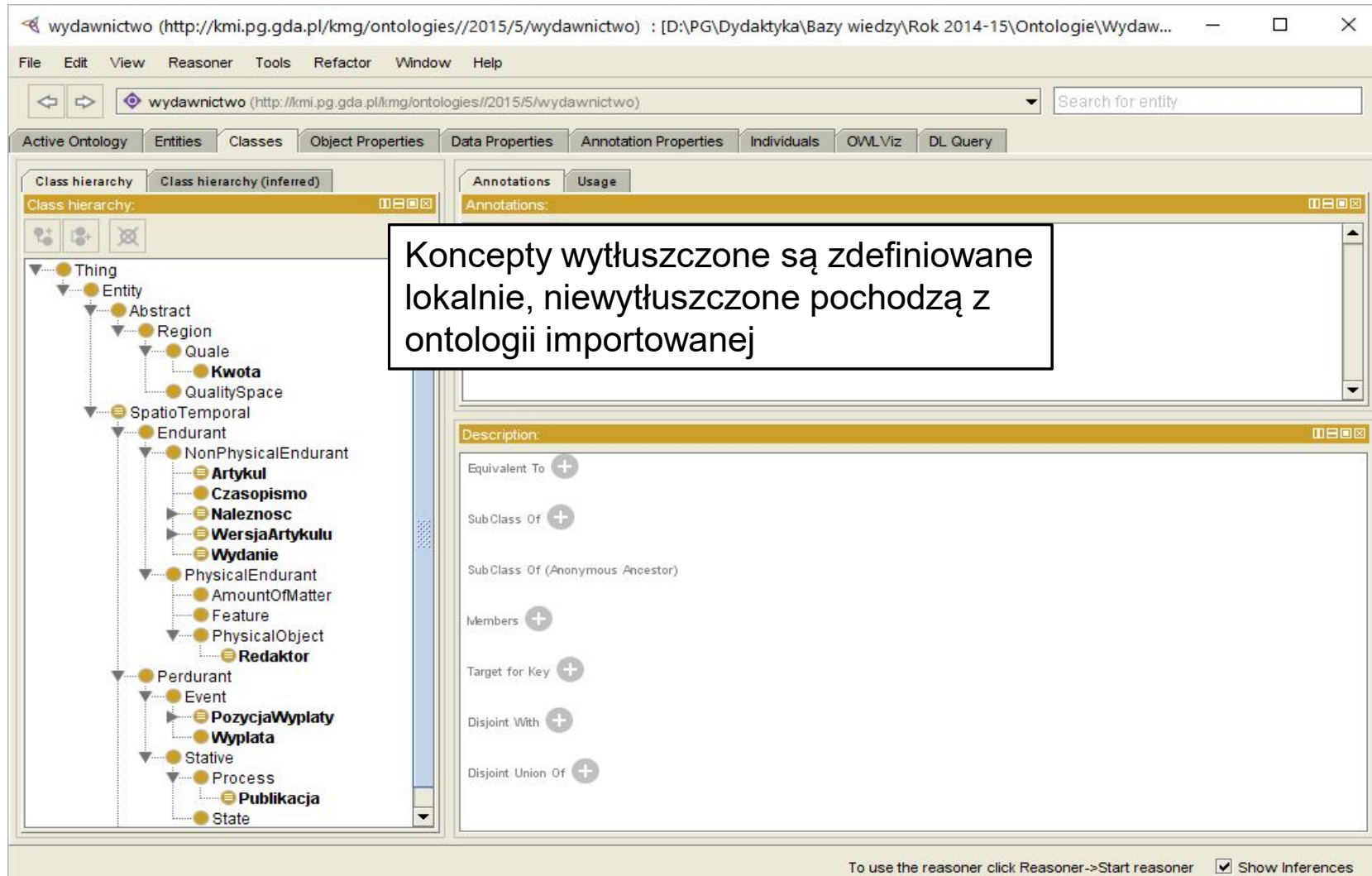
Attributive language. This is the base language which allows:

- Atomic negation (negation of concepts that do not appear
- Concept intersection
- Universal restrictions

Rules +

To use the reasoner click Reasoner->Start reasoner Show Inferences

Powtórne użycie



wydawnictwo (http://kmi.pg.gda.pl/kmg/ontologies//2015/5/wydawnictwo) : [D:\PG\Dydaktyka\Bazy wiedzy\Rok 2014-15\Ontologie\Wydaw...

File Edit View Reasoner Tools Refactor Window Help

wydawnictwo (http://kmi.pg.gda.pl/kmg/ontologies//2015/5/wydawnictwo) Search for entity

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWL Viz DL Query

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy:

- Thing
 - Entity
 - Abstract
 - Region
 - Quale
 - Kwota**
 - QualitySpace
 - SpatioTemporal
 - Endurant
 - NonPhysicalEndurant
 - Artykul**
 - Czasopismo**
 - Nalezosc**
 - WersjaArtykulu**
 - Wydanie**
 - PhysicalEndurant
 - AmountOfMatter
 - Feature
 - PhysicalObject
 - Redaktor**
 - Perdurant
 - Event
 - PozycjaWyplaty**
 - Wyplata**
 - Stative
 - Process
 - Publikacja**
 - State

Annotations:

Annotations:

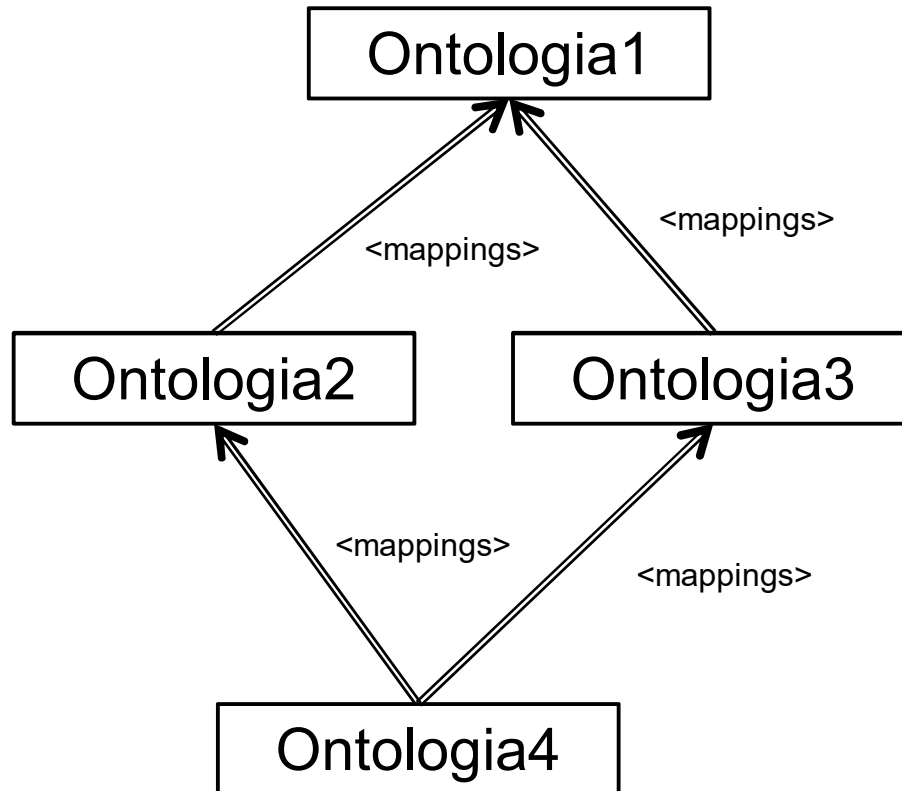
Description:

- Equivalent To +
- Sub Class Of +
- Sub Class Of (Anonymous Ancestor)
- Members +
- Target for Key +
- Disjoint With +
- Disjoint Union Of +

To use the reasoner click: Reasoner -> Start reasoner Show Inferences

Koncepty wytłuszczone są zdefiniowane lokalnie, niewytłuszczone pochodzą z ontologii importowanej

Powtórne użycie

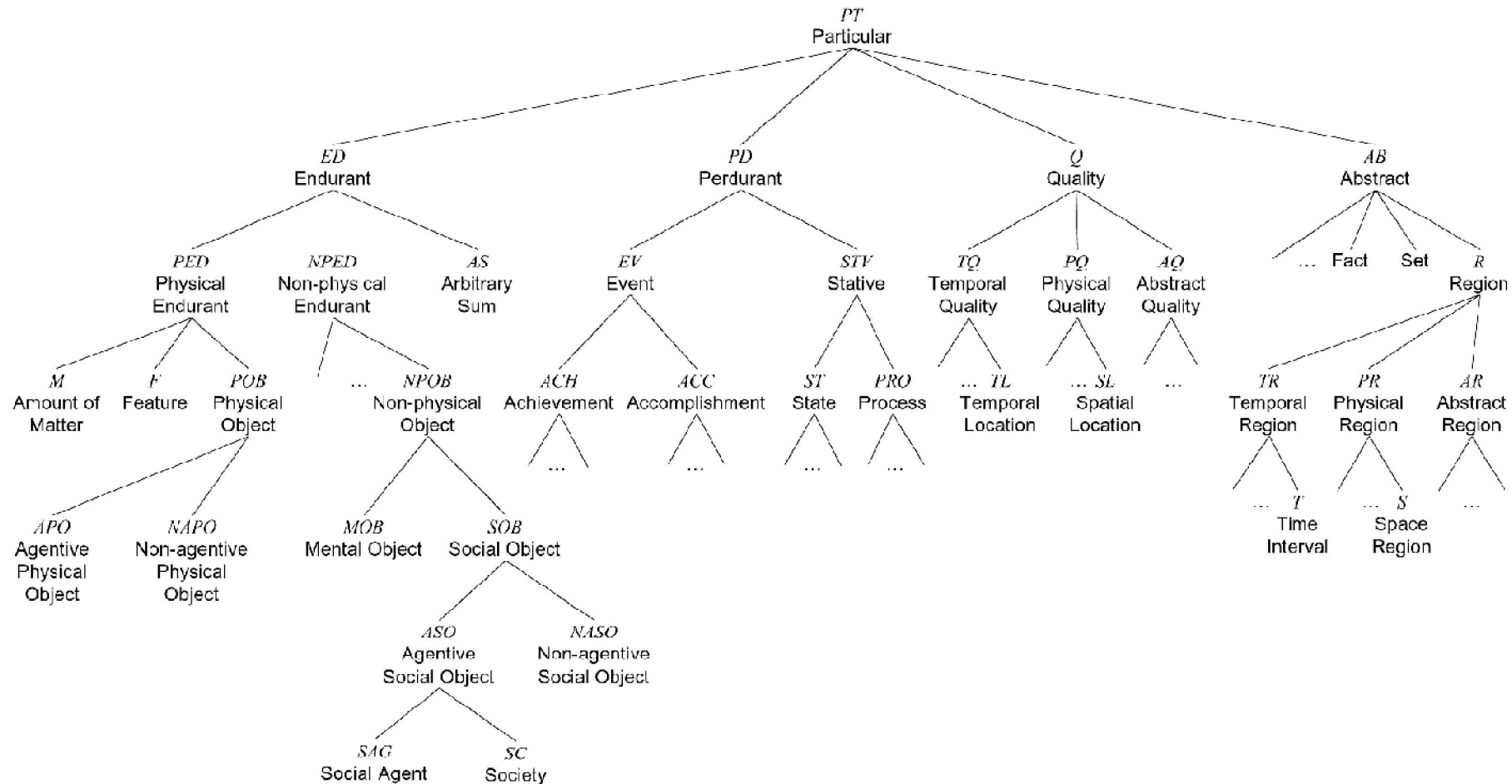


Proponowane są także techniki pozwalające na powtórne użycie tylko części ontologii. Nazywane są uzgadnianiem ontologii (ang. *ontology alignment*) lub zrównywaniem ontologii (ang. *ontology matching*). Wiążą się one z koniecznością opracowywania reguł przenoszenia wniosków między ontologiami, nazywanych odwzorowaniami (ang. *mappings*).



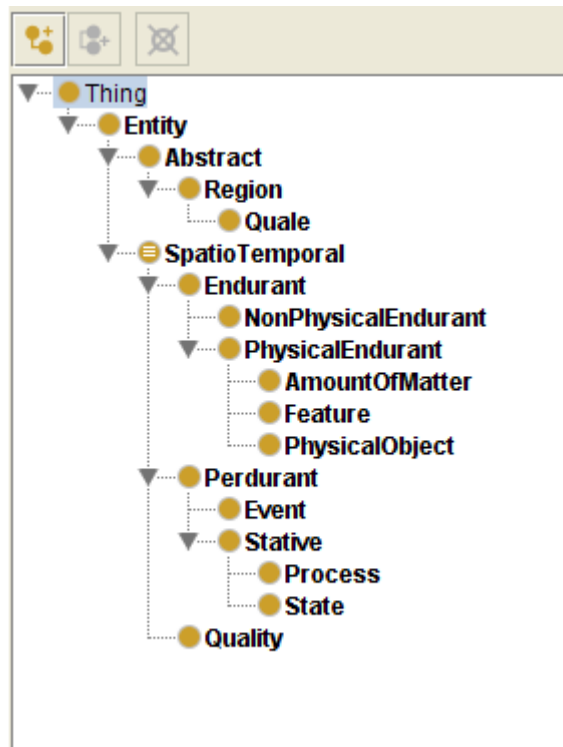
DOLCE, SweetEssence

W naszych zajęciach będziemy wykorzystywali ontologię DOLCE, a właściwie jej fragment nazwany Sweet Essence.

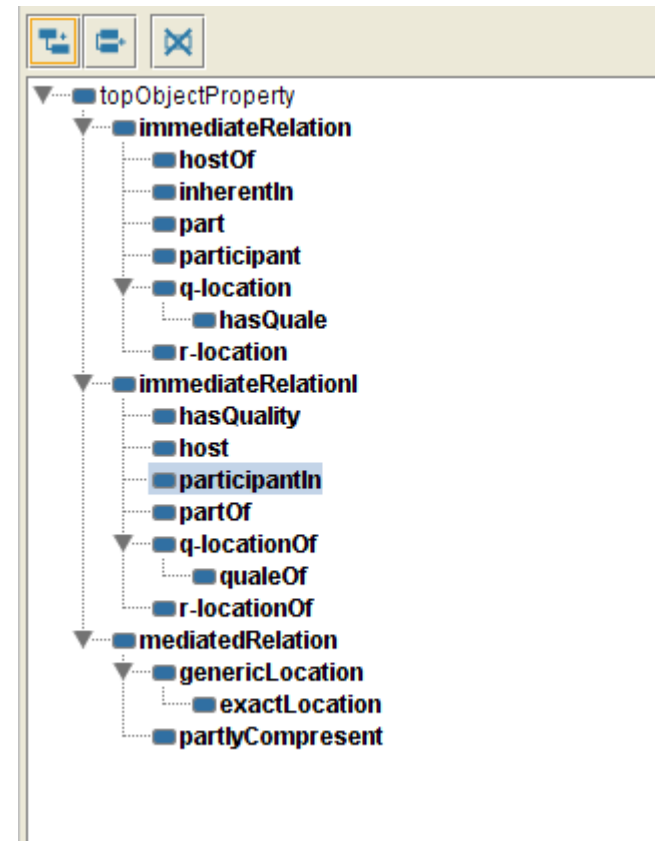


Fragment ontologii DOLCE wyodrębniony na potrzeby naszych zajęć:

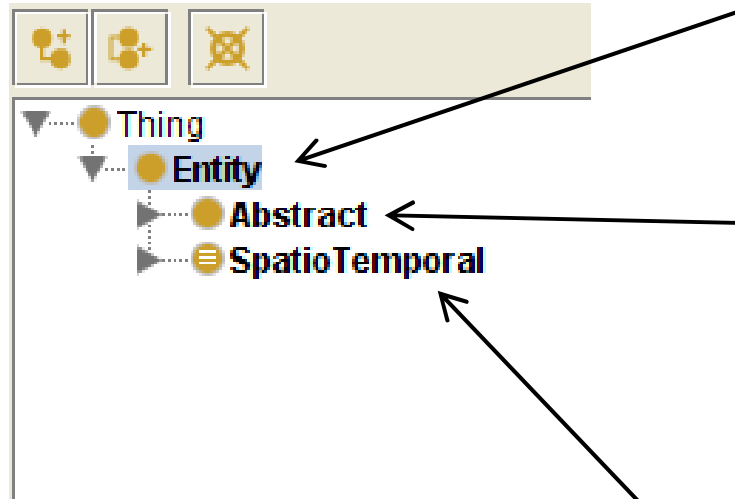
Koncepty:



Role:



SweetEssence

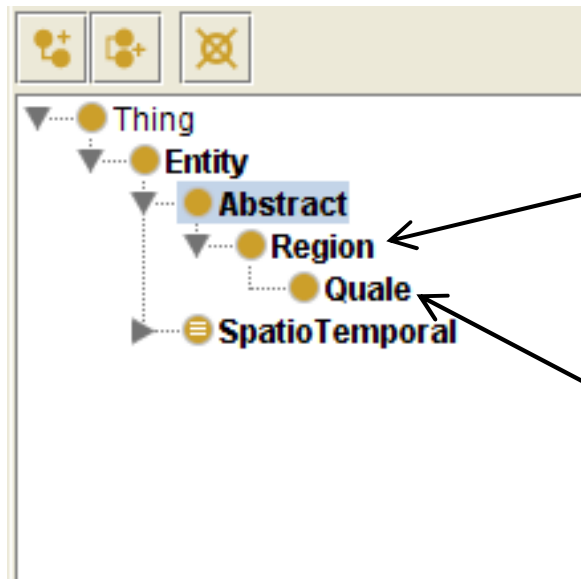


Entity: Wszystkie osobniki w świecie ontologii, o których możemy rozmawiać (w oryg. DOLCE nazywa się Particular).

Abstract: Obiekty abstrakcyjne nie mające własności czasowych ani przestrzennych, np. czerwień; same nie są też własnościami (lecz ich wartościami).

SpatioTemporal: Byty, które są czaso-przestrzenne.

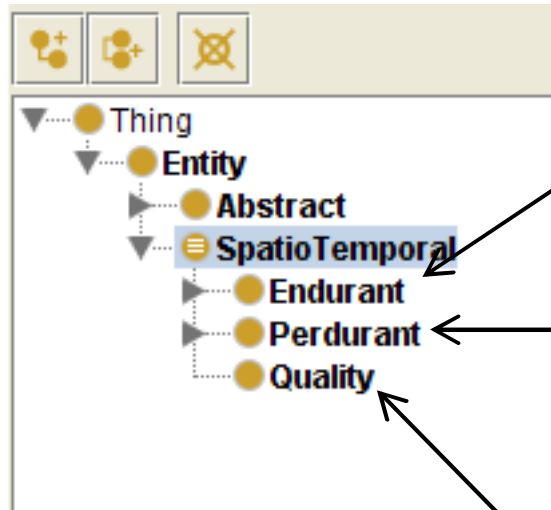
SweetEssence



Region: podzbiór przestrzeni qualiów, który może przekładać się np. na obszar oznaczający kolor czerwony w ogóle (obejmujący wszystkie odcienie czerwonego).

Quale (l.mn. qualia): świadome doznanie jakościowe związane z bodźcami zmysłowymi (np. doznanie koloru czerwonego). Prościej: atomowy (niepodzielny) obszar w przestrzeni wartości własności.

SweetEssence

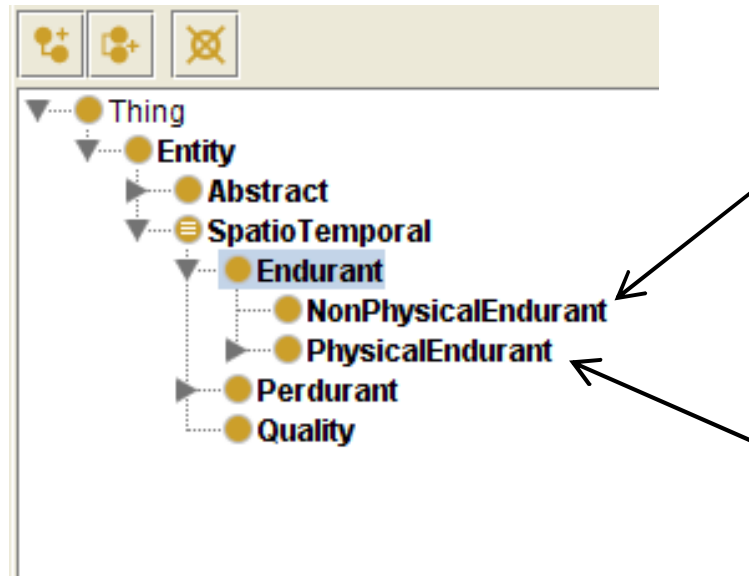


Endurant: Byt, który jest swoistą całością w każdej pojedynczej (najmniejszej możliwej do wyróżnienia) chwili czasu. (byt trwający w czasie).

Perdurant: Byt odbywający się w czasie (stan, proces, zdarzenie). Cechą charakterystyczną tego bytu, jest to, że w pojedynczej (najmniejszej możliwej do wyróżnienia) chwili czasu "odbywa się" zawsze jedynie fragment tego bytu (np. część "czytania"). (byt rozciągły w czasie).

Quality: Własność konkretna, którą możemy postrzegać i/lub mierzyć. Na przykład "kolor konkretnej róży".

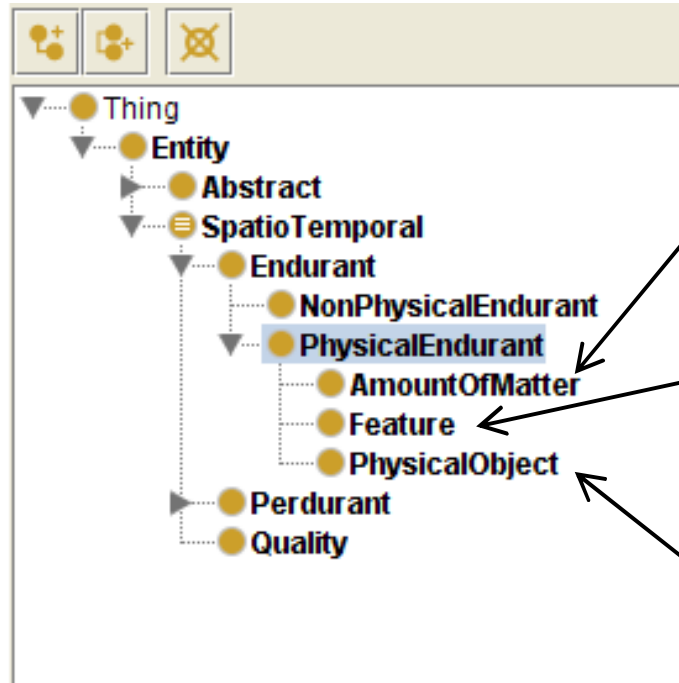
SweetEssence



NonPhysicalEndurant: Byt trwający w czasie nie posiadający (bezpośrednio) własności przestrzennych zależny od pewnego agenta.

PhysicalEndurant: Byt posiadający (w sposób bezpośredni) pewne fizyczne lub przynajmniej przestrzenne własności.

SweetEssence

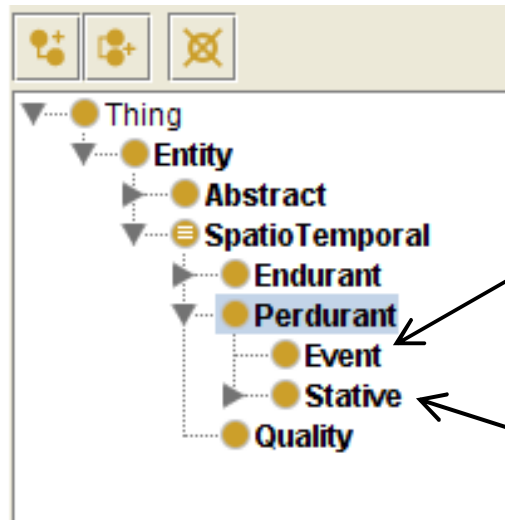


AmountOfMatter: Byt trwający w czasie nie utrzymujący całości: np. złoto, cukier.

Feature: Byt, którego istnienie zależne jest od jego bytu "nadrzędnego", np. krawędź blatu stołu, dziura w serze, przestrzeń pod łóżkiem itp.

PhysicalObject: Fizyczny obiekt, tj. byt trwający w czasie o cechach przestrzennych, posiadający własne kryterium całości (choć różne dla różnych obiektów).

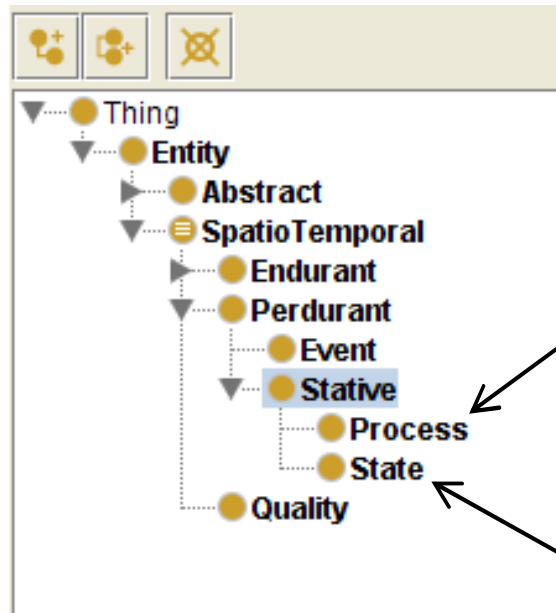
SweetEssence



Event: Zdarzenie. Byt rozciągły w czasie, który oznacza przejście od stanu do stanu (zwracamy tu uwagę na skutek [zdarzenia]).

Stative: Byt rozciągły w czasie, który charakteryzuje się tym, że jeśli złączyć (zsumować mereologicznie) dwa podobne byty otrzymamy byt tego samego rodzaju. Na przykład siedzenie [w fotelu] + siedzenie [w fotelu] = (dłuższe) siedzenie [w fotelu]. ("statyczny" byt rozciągły w czasie).

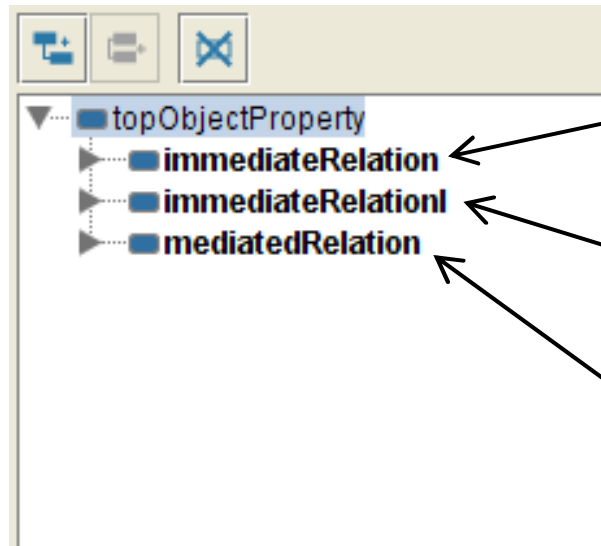
SweetEssence



Process: Niejednolity "statyczny" byt rozciągnięty w czasie; np. lepienie pierogów, którego charakterystyczną cechą jest to, że składa się z kolejnych następujących po sobie faz: np. nabieranie farszu, nakładanie ciasta, wycinanie, zalepianie.

State: Jednolity "statyczny" byt rozciągnięty w czasie; np. siedzenie, którego charakterystyczną cechą jest to, że w każdej chwili "siedzimy".

SweetEssence



immediateRelation

Relacja, która zachodzi bezpośrednio między dwoma osobnikami (niezłożona).

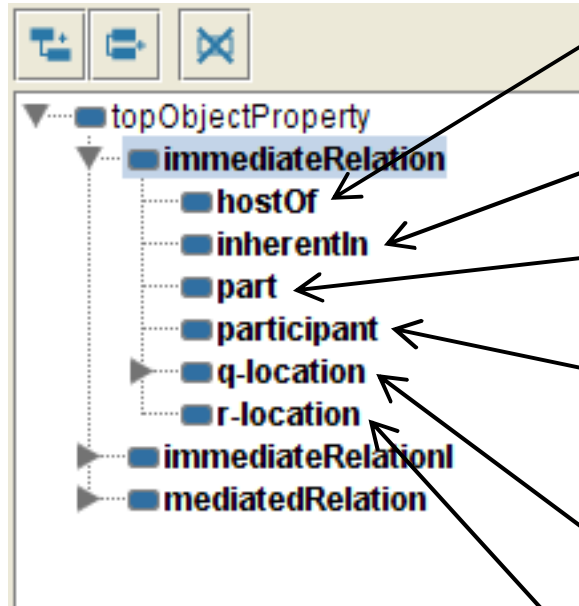
immediateRelationI

Relacje odwrotne do wyróżnionych relacji bezpośrednich.

mediatedRelation

Relacje, które "omijają" część informacji, aby np. połączyć bezpośrednio dwa byty trwające w czasie (z pominięciem bytu rozciągniętego w czasie).

SweetEssence



hostOf (D: PhysicalEndurant , Z: Feature)
Relacja posiadania encji "podrzędnych"
(wskazująca na encje "podrzędne", nie mogące
istnieć bez naszej encji).

inherentIn (D: Quality, Z: Entity)
Relacja bycia własnością encji.

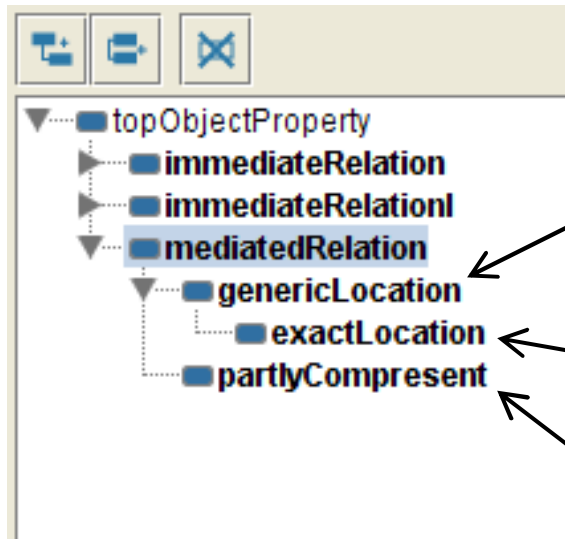
part (D: Entity, Z: Entity)
Relacja posiadania części.

participant (D: Perdurant, Z: Endurant)
Relacja partycypacji łącząca byt rozciągły w
czasie z uczestniczącymi w nim (tj. w tym
wydarzeniu, procesie itp.) bytami trwającymi w
czasie.

q-location (D: Quality, Z: Region)
Relacja posiadania wartości (przedziału
wartości) własności.

r-location (D: Region, Z: Region)
Relacja posiadania podregionów (wskazująca na
regiony zawarte w naszym regionie).

SweetEssence



genericLocation (D: Entity, Z: Entity)
Relacja wyrażająca wartość pewnej (nieznanej) własności encji, niekoniecznie wprost.

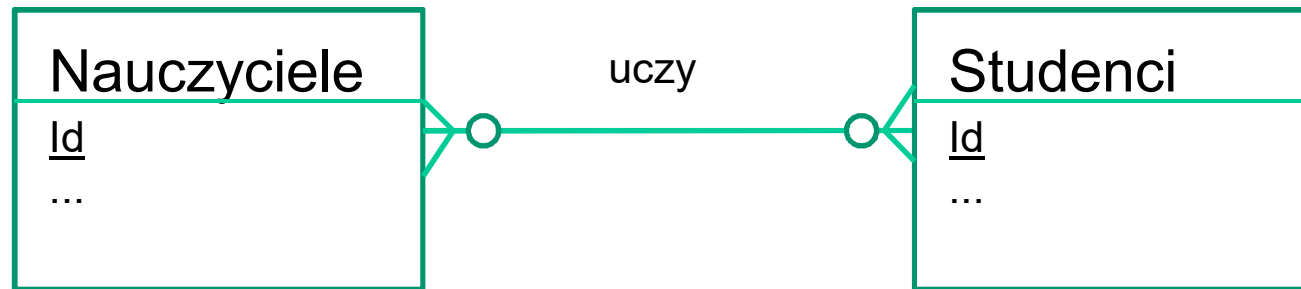
exactLocation (D: Entity, Z: Region)
Relacja wyrażająca wprost wartość pewnej (nieznanej) własności encji.

partlyCompresent (D: Endurant, Z: Endurant)
Relacja pomiędzy bytami trwającymi w czasie, oznaczająca, że musi istnieć przynajmniej jeden moment czasowy, w którym byty te współistnieją.

Niektóre inne ograniczenia:

1. Koncepty *Abstract* i *SpatioTemporal* są rozłączne.
2. Koncepty *Endurant*, *Perdurant* i *Quality* są wzajemnie rozłączne.
3. Rola *part*, choć jej dziedziną i zakresem jest koncept *Entity*, może łączyć tylko „endurant” z „endurancem” lub „perdurant” z „perdurancem”.
4. Rola *participant* może łączyć tylko „perdurant” z „endurancem”.

SweetEssence – przykład



(Prawie) najprostsze dostosowanie do Dolce:

Nauczyciel \sqsubseteq Endurant

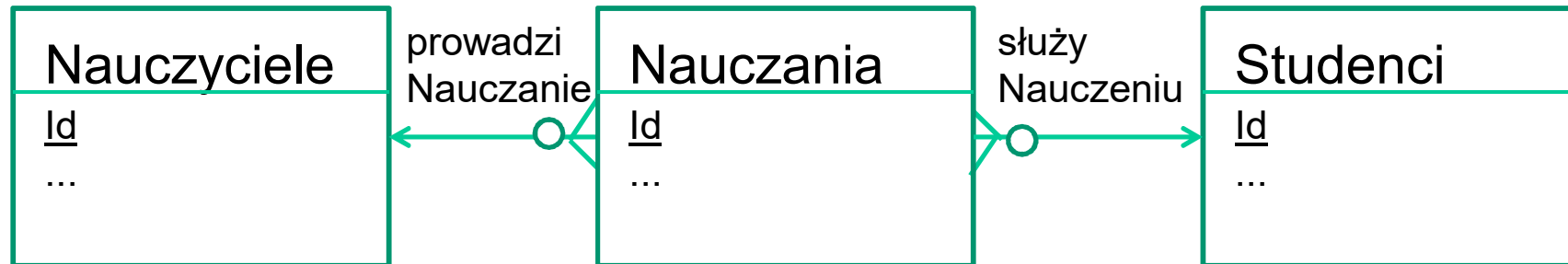
Student \sqsubseteq Endurant

uczy \sqsubseteq mediatedRelation

(istnieje coś skomplikowanego między
nauczycielem a studentem)

SweetEssence – przykład

Lepsze dostosowanie do Dolce – zmiana modelu:



Nauczyciel \sqsubseteq Endurant

Student \sqsubseteq Endurant

Nauczanie \sqsubseteq Perdurant

prowadziNauczanie \sqsubseteq participantIn

służyNauczeniu \sqsubseteq participant

(uczy \sqsubseteq mediatedRelation = prowadziNauczanie o służyNauczeniu)

SweetEssence – przykład

Dokładniejsze dostosowanie do Dolce:

dla „endurantów” podstawowe rozróżnienie to
PhysicalObject i NonPhysicalEndurant:

Nauczyciel (i Student) może być rozumiany jako PhysicalObject. To najprostsze podejście, umożliwia formułowanie zdań typu: *Adam Nowak jest Nauczycielem.*

Nauczyciel (i Student) może być rozumiany jako NonPhysicalEndurant. Jest rozumiany wówczas jako rola społeczna. Umożliwia formułowane zdań typu *Adam Nowak pełnił rolę Nauczyciela w latach 2005-2007.*

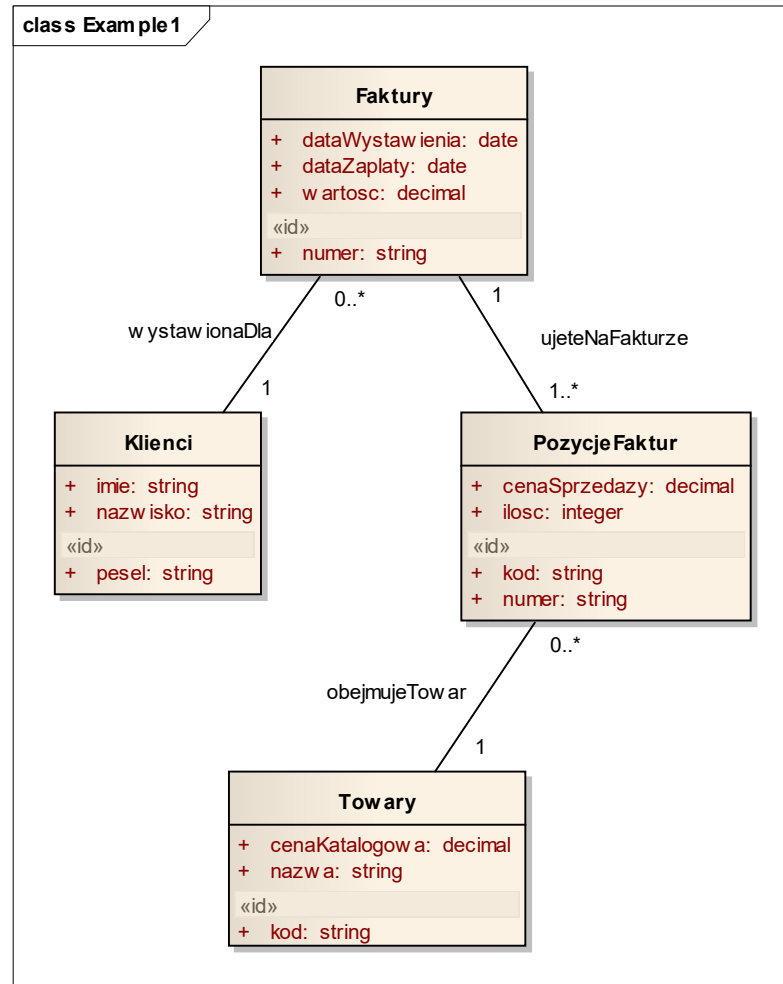
SweetEssence – przykład

Dokładniejsze dostosowanie do Dolce:

dla „perdurantów” podstawowe rozróżnienie to
Event, Process i State:

Nauczanie powinno być rozumiane dość jednoznacznie jako Process. Event to bardziej punkt w czasie, a nauczanie rozciąga się w czasie. State to coś, co rozciąga się w czasie, ale jest statyczne, a (jak wszyscy wiemy) nauczanie jest dynamiczne, dużo się różnych rzeczy w jego trawieniu dzieje (np. kolokwia) i może być dość burzliwe.

ERD:

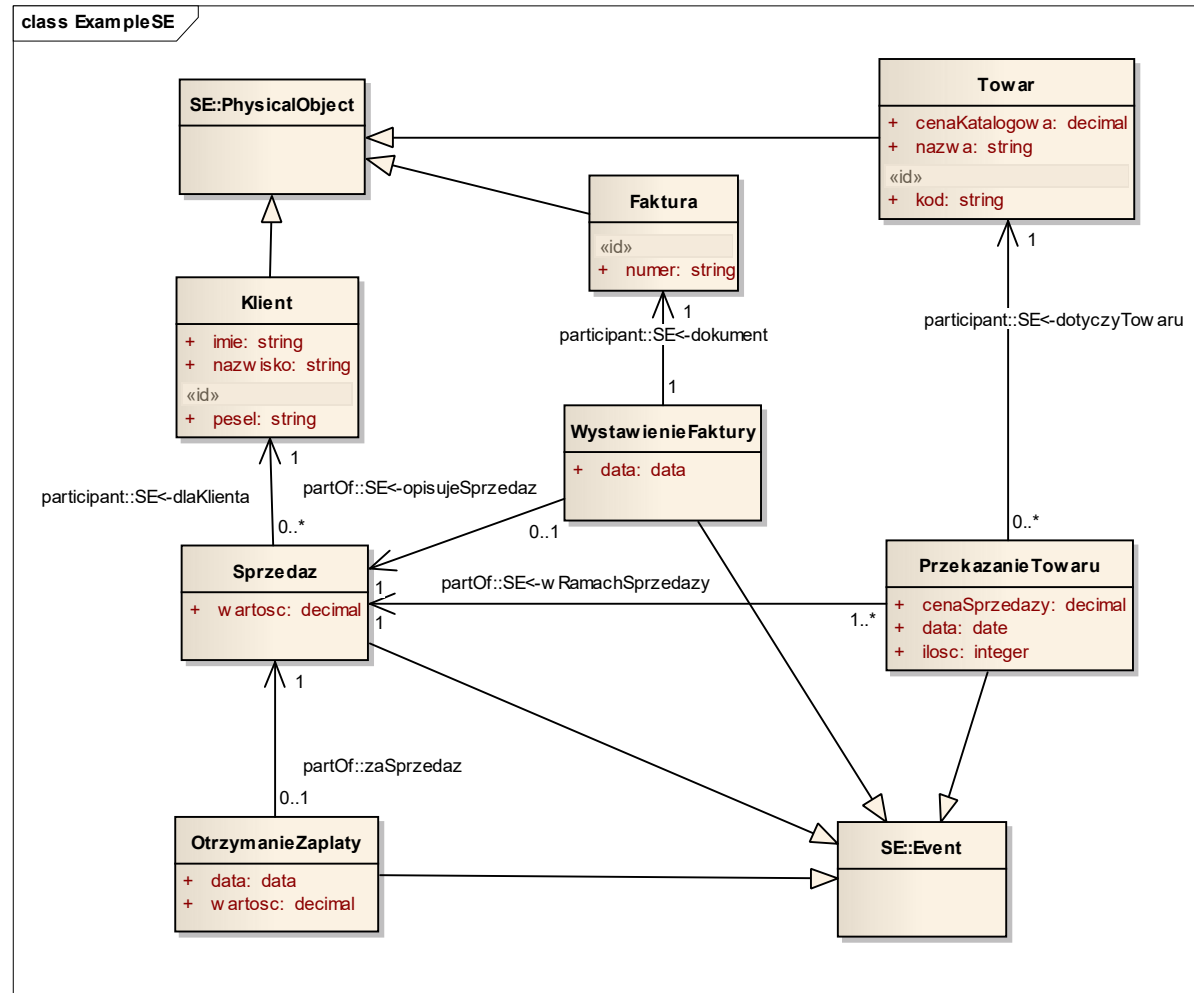


Standardowy model sprzedaży z projektów studenckich:

Zbiory encji Faktury i PozycjeFaktur reprezentują nie tylko dokument, ale także szereg zdarzeń: przekazanie towarów, wystawienie dokumentu, uiszczenie zapłaty.



Ontologia:



W ontologii korzystającej z SweetEssence (a także z Dolce) powinniśmy rozbić sprzedaż na kilka odpowiednich zdarzeń, a fakturę rozumianą jako dokument wyodrębnić jako obiekt fizyczny.