

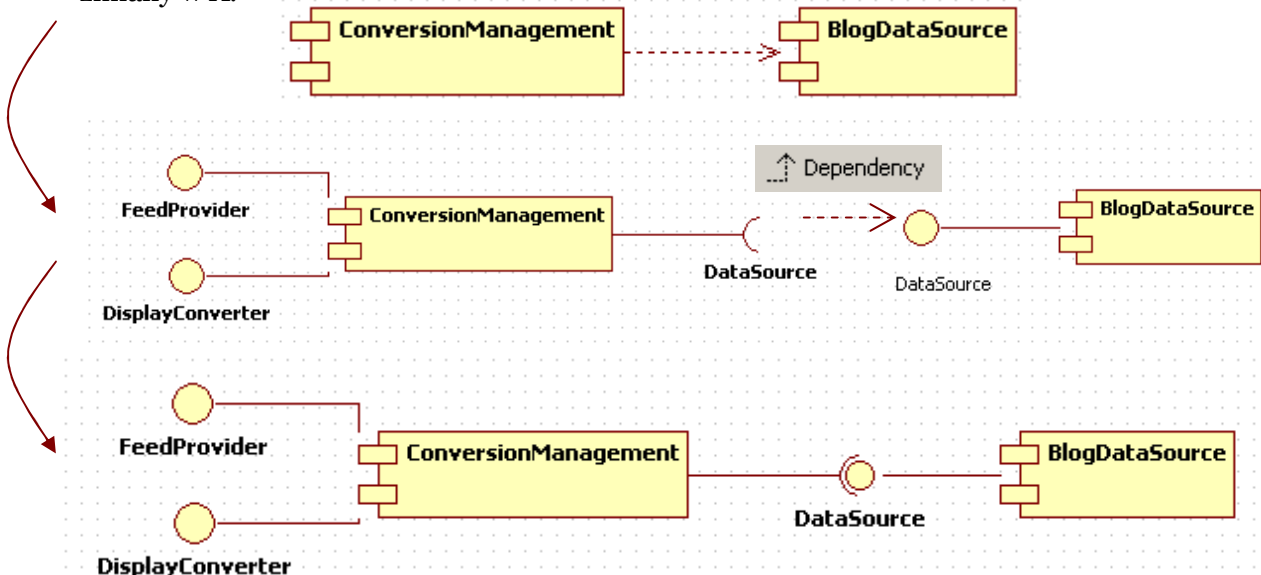
DIAGRAMY KOMPONENTÓW I PAKIETÓW

1) Zastosowanie

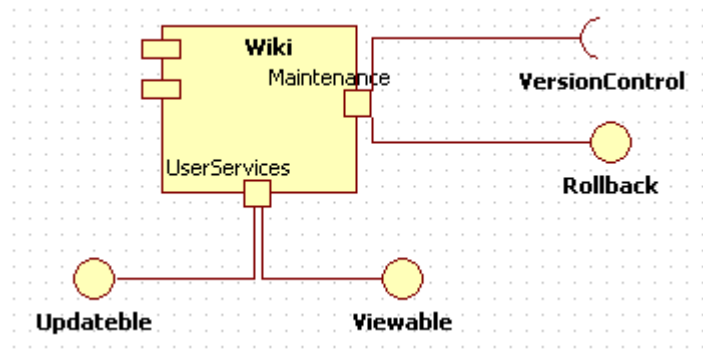
- ▶ Diagram komponentów i pakietów w UML 2.0 możemy zrealizować w StarUML w **Development view** | **Component Diagram**. Oba te diagramy składają się na model implementacji systemu.
- ▶ **Development view** ma na celu zilustrowanie systemu z perspektywy programisty, w centrum zainteresowania znajduje się zarządzanie oprogramowaniem.
- ▶ Diagramy te nie mają na celu przedstawienia w pełni szczegółowego systemu, a jedynie ideę. Pomocniczą techniką wyboru najważniejszych aspektów jest zasada „jednej minuty w windzie” – gdy spotkasz w windzie prezesa, lub potencjalnego klienta masz minutę na zaprezentowanie wizji systemu, co powiesz?
- ▶ **Komponent** reprezentuje modułowy, rozwijający się, oraz zamienny element systemu, który zawiera się w implementacji oraz ma kontakt ze zbiorem interfejsów. Reprezentuje podsystem (grupę powiązanych klas), który wymaga pewnych interfejsów, a pewne interfejsy implementuje.

2) Diagram komponentów

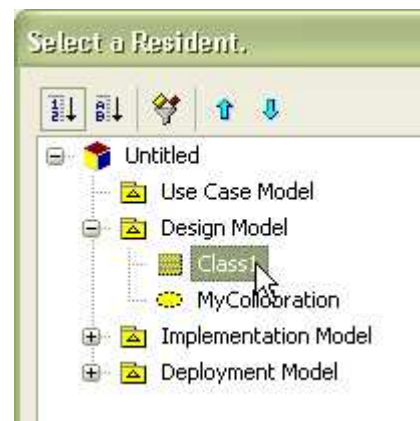
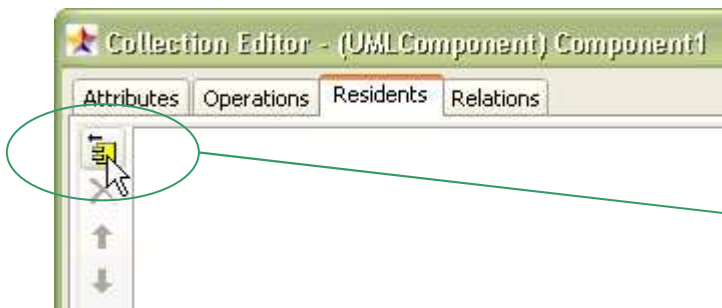
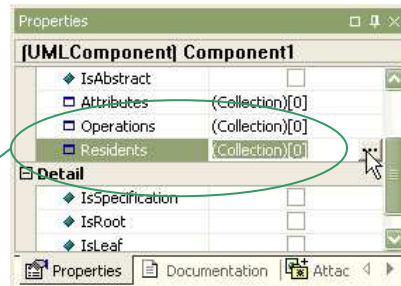
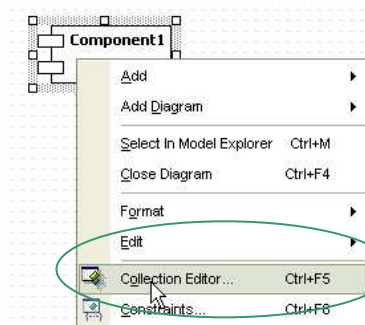
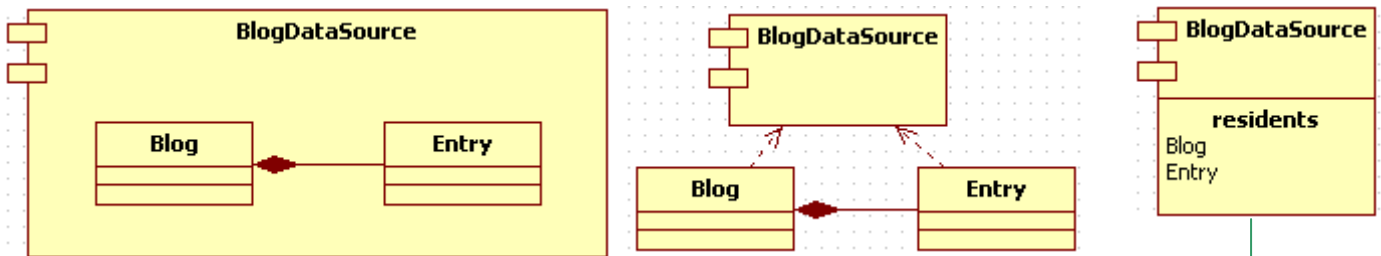
- ▶ **Diagram komponentów** pokazuje zależności pomiędzy komponentami i interfejsami
- ▶ **Komponent** - reprezentuje jeden fizyczny moduł kodu. Często jest to jeden pakiet, ale nie zawsze istnieje taka jednoznaczna odpowiedniość. Można je grupować w pakiety.
- ▶ Zdefiniowane są następujące **stereotypy komponentów**:
 - executable - określa komponent, który można wykonać na węźle;
 - library - określa dynamiczną lub statyczną bibliotekę obiektów;
 - table - określa komponent reprezentujący tabelę bazy danych;
 - file - komponent reprezentujący dokument zawierający kod źródłowy lub dane;
 - document - określa komponent reprezentujący dokument;
- ▶ Komponenty są między sobą powiązane relacją zależności, ponieważ wymagają ich do realizacji własnej funkcjonalności. Zależność między A i B oznacza, że komponent A korzysta z komponentu B i zmiana w komponencie B może spowodować konieczność zmiany w A.



- Port jest punktem interakcji pomiędzy komponentem, a otoczeniem. Wyznacza różne sposoby użycia komponentu przez świat zewnętrzny. Porty są związane z interfejsami. Port może grupować interfejsy aby pokazać funkcjonalności udostępniane przez port.

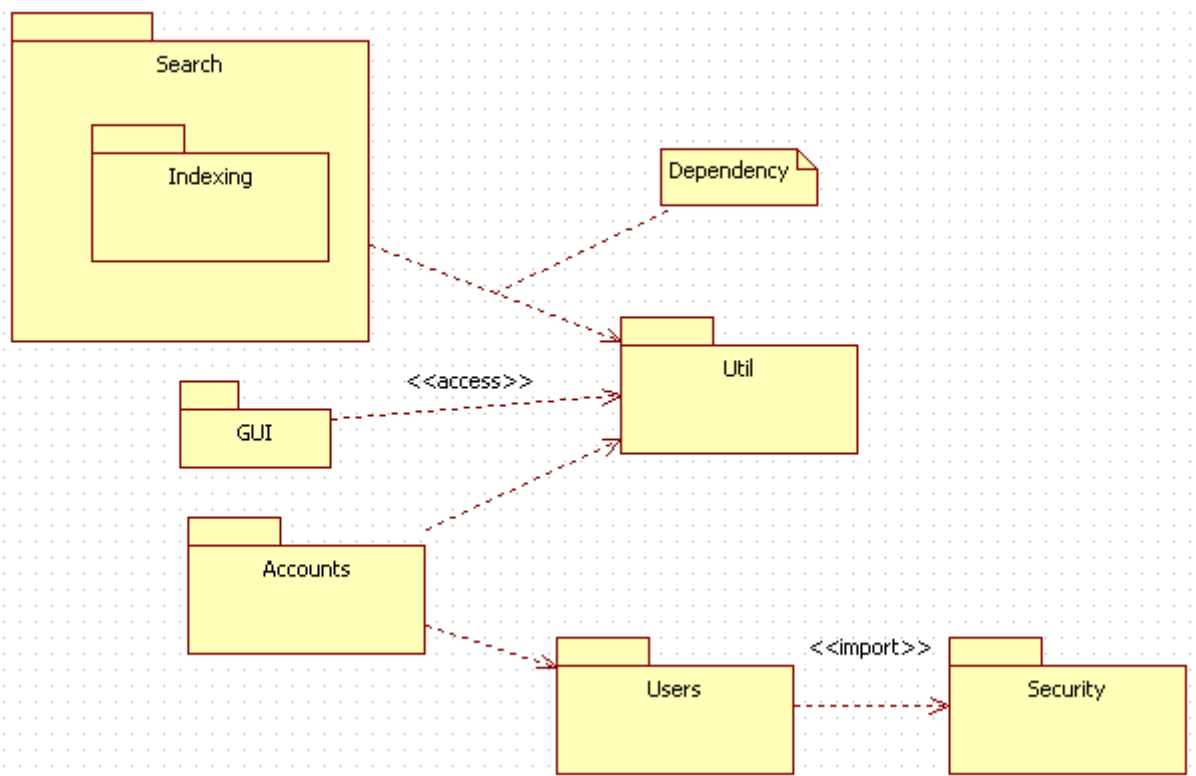


- Komponenty często zawierają klasy, które implementują ich funkcjonalności. Możemy je rysować wewnątrz komponentu lub jako klasy zależne na zewnątrz, lub jedynie wymieniać jako gościa (*resident*)



3) Diagram pakietów

- ▶ Pakiet jest zbiorem elementów modelu. Pakiety mogą być zagnieżdżone w innych pakietach. Pakiet może zawierać podporządkowane pakiety ale także i inne elementy modelu, przede wszystkim klasy. Wszystkie elementy modelu UML mogą być grupowane w pakiety według dowolnego kryterium.
- ▶ Pakiety doskonale nadają się do wizualizacji podstawowych **zależności pomiędzy częściami systemu**, dzięki czemu łatwo ocenić jakość i stopień powiązań pomiędzy nimi. Dobra struktura pakietów, w której zależności są jasno uporządkowane oraz nie występują (lub występują tylko na niskim poziomie) zależności cykliczne, wspiera późniejszą rozbudowę systemu. W szczególności przydają się w dużych aplikacjach, podzielonych na wiele podsystemów, ponieważ w prosty sposób obrazują podstawowe zależności pomiędzy nimi.
- ▶ Pakiet tworzy także jednostkę hermetyzacji: elementy z pakietu odwołują się do elementów zewnętrznych posługując się ich pełnymi kwalifikowanymi (zawierającymi nazwy pakietów) nazwami, zgodnie z ich zakresem widoczności, natomiast wewnątrz pakietu elementy mogą odwoływać się do siebie bezpośrednio.



4) Źródła:

- Larman C., Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, Prentice Hall PTR Upper Saddle River, 2001
- Russ, Hamilton, Learning UML 2.0, O'Reilly, 2006
- Uczelnia OnLine, Inżynieria oprogramowania, prowadzący: Walter B., UML cz. II, <http://wazniak.mimuw.edu.pl/>
- Wolski M., StarUML Rozdział 5.7. Modelowanie przy użyciu diagramu komponentów, <http://www.michalwolski.com/2008/02/staruml-rozdzial-57-modelowanie-przy-uzyciu-diagramu-komponentow/>