Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Informatică și Ingineria Sistemelor

**RAPORT**

Lucrare de laborator Nr.1

Disciplina: APSI

Tema: Familiarizarea cu instrumentul CASE „Enterprise Architect” şi analiza generală a principiilor de modelare în baza limbajului de modelare UML. Studierea şi descrierea destinaţiei funcţionale a submeniurilor/opţiunilor din meniuri

A efectuat: st.gr.

Ulmanu Cristian IA-182,

A verificat : lect.univ.   
 Sava Nina  
Melnic Radu

Chișinău 2020

**Scopul**: studierea elementelor și entităților instrumentului de modelare Enterprise Architect.  
 **Sarcina:** de realizat o descriere a elementelor de bază Enterprise Architect

**Considerații teoretice**

**Unified Modeling Language** (prescurtat **UML**) este un limbaj standard pentru descrierea de [modele](https://ro.wikipedia.org/w/index.php?title=Modelare_orientat%C4%83_pe_obiect&action=edit&redlink=1" \o "Modelare orientată pe obiect — pagină inexistentă) și [specificații](https://ro.wikipedia.org/w/index.php?title=Specifica%C8%9Bie&action=edit&redlink=1" \o "Specificație — pagină inexistentă) pentru [software](https://ro.wikipedia.org/wiki/Software). Limbajul a fost creat de către consorțiul [Object Management Group](https://ro.wikipedia.org/w/index.php?title=Object_Management_Group&action=edit&redlink=1) (OMG) care a mai produs printre altele și standardul de schimb de mesaje intre sisteme [CORBA](https://ro.wikipedia.org/w/index.php?title=CORBA&action=edit&redlink=1). UML a fost la bază dezvoltat pentru reprezentarea complexității programelor orientate pe obiect, al căror fundament este structurarea programelor pe clase, și instanțele acestora (numite și obiecte). Cu toate acestea, datorită eficienței și clarității în reprezentarea unor elemente abstracte, UML este utilizat dincolo de domeniul IT. Așa se face că există aplicații ale UML-ului pentru management de proiecte, pentru *business Process Design* etc.

Prima versiune de UML, UML 1.0, a apărut în anul 1990 ca reacție a numeroaselor limbaje de modelare propuse pe piață. UML îi are ca fondatori pe [Grady Booch](https://ro.wikipedia.org/w/index.php?title=Grady_Booch&action=edit&redlink=1), [Ivar Jacobson](https://ro.wikipedia.org/w/index.php?title=Ivar_Jacobson&action=edit&redlink=1) și [James Rumbaugh](https://ro.wikipedia.org/w/index.php?title=James_Rumbaugh&action=edit&redlink=1), așa numiții „cei trei *Amigos*”. Ei au dezvoltat limbajul bazându-se inclusiv pe limbaje de modelare deja existente, însă incomplete ca gamă de funcționalități. Printre acestea se numără și OOSE, RDD, OMT, OBA, OODA, SOMA, MOSES și OPEN/OML.

### UML 2.x

Dezvoltarea versiunii 2 a UML a început în anul 1999 atunci când OMG a publicat un [*request for information*](https://ro.wikipedia.org/w/index.php?title=Request_for_information&action=edit&redlink=1) referitor la UML 2. De atunci, UML s-a aflat într-un continuu ciclu de îmbunătățire, astăzi ajungând la varianta UML 2.4.1 (publicată în august 2011).

Sparx Systems Enterprise Architect este un instrument de modelare și proiectare vizuală bazat pe OMG UML. Platforma suportă: proiectarea și construirea sistemelor software; modelarea proceselor de afaceri; și modelarea domeniilor bazate pe industrie. Este folosit de companii și organizații pentru a nu doar modela arhitectura sistemelor lor, ci și pentru a procesa implementarea acestor modele pe parcursul întregului ciclu de viață al dezvoltării aplicațiilor.

**Implementare, rezultate practice:**

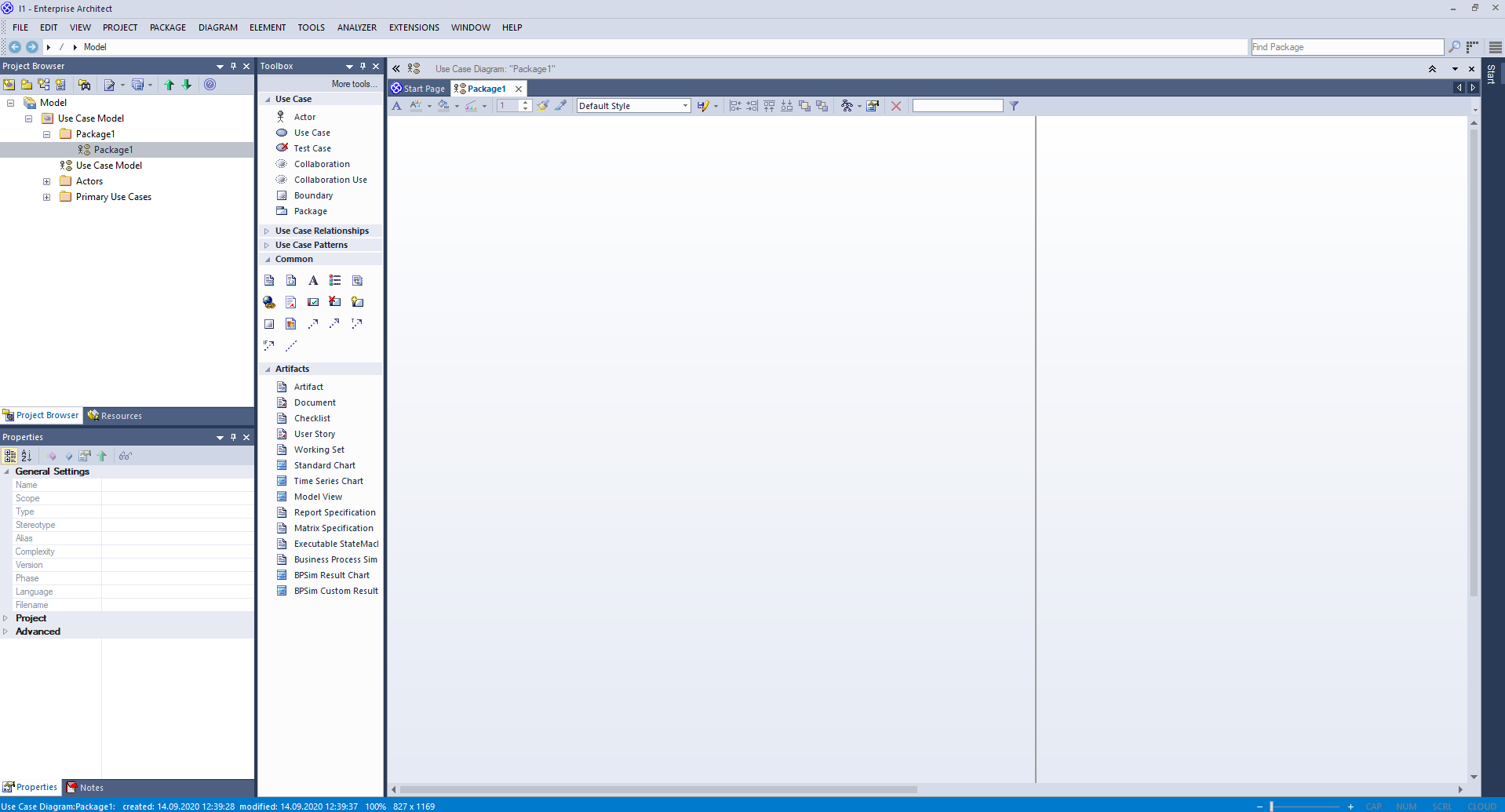


Figure 1 Interfata Enterprise Architect

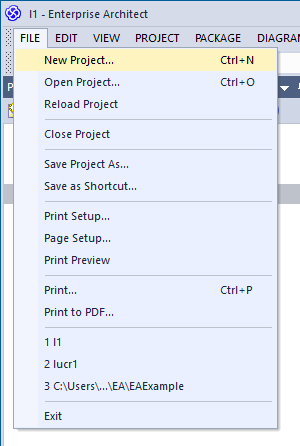


Figure 2 Crearea unui nou proiect

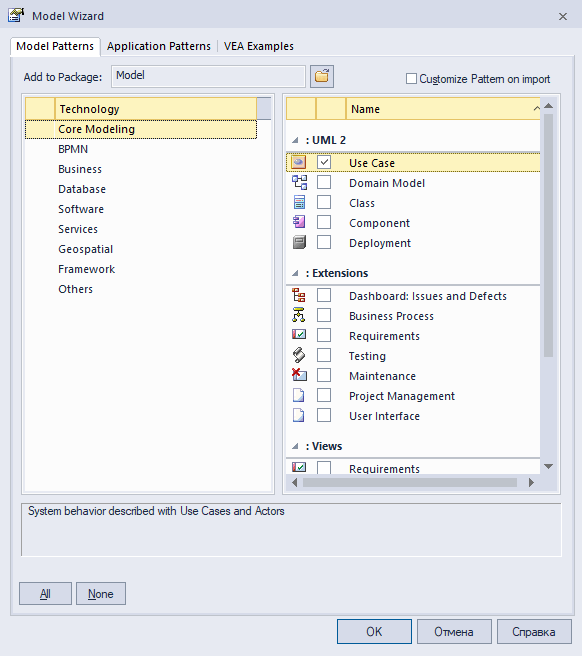


Figure 3 Alegem tip de UML si tehnologia

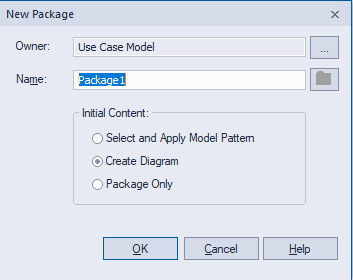


Figure 4 Cream un nou package

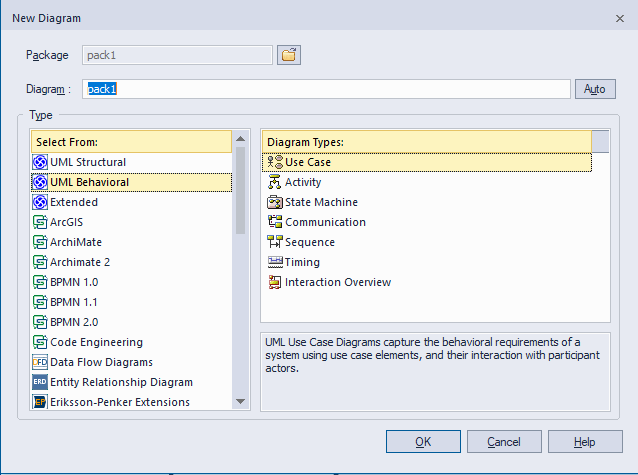


Figure 5 Alegem ce tip de diagrama vom crea

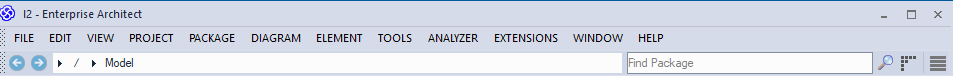


Figure 6 Bara standarda de meniuri

Default Tools Toolbar- oferă acces instantaneu la instrumentele cele mai folosite în Enterprise Architect.

***Se utilizează la:***

* Creare proiecte noi și deschidere proiecte existente
* Salvarea modificărilor la diagrama curentă
* Editare, tăiere, copiere și lipirea obiectelor din diagrame
* Accesare Ajutor Enterprise Architect
* Editarea proprietăților la fiecare element

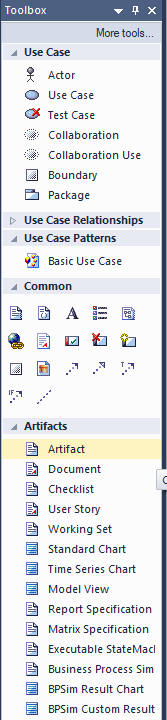


Figure 7 Toolbox

* În toolbox putem adăuga actori, use case, test case, etc.
* Putem alege ce case relationships sa alegem.
* Putem adăuga notițe, legenda diagramei, hyperlink, etc.
* În More tools putem schimba tipul de diagramă.

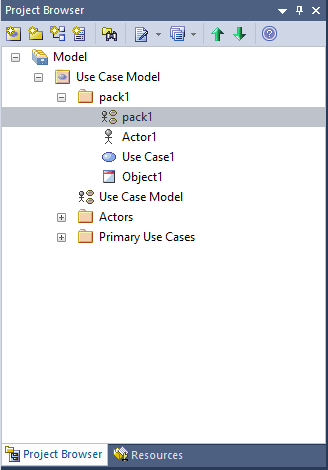


Figure 8 Project browser

* În project browser avem structura proiectului care constă din package și în cadrul la package avem elementele de pe diagramă.

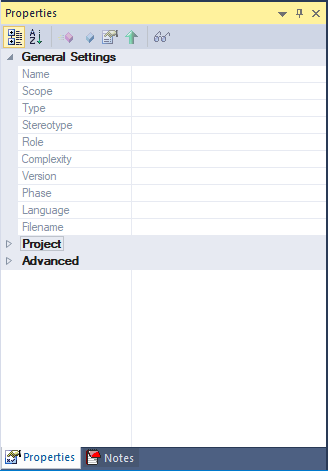


Figure 9 Fereastra properties și notes

* Putem observa și modifica fiecare proprietate a elementului din diagramă.

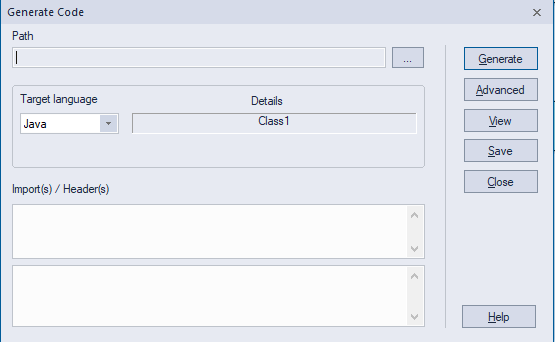


Figure 10 Generare de cod

* Pentru a genera codul accesăm more tools din toolbox și alegem class. După ce adăugăm class putem apăsa tasta dreaptă de la mouse și să alegem Code Engineering și alegem Generate Code și alegem limbajul codului și ce importăm.

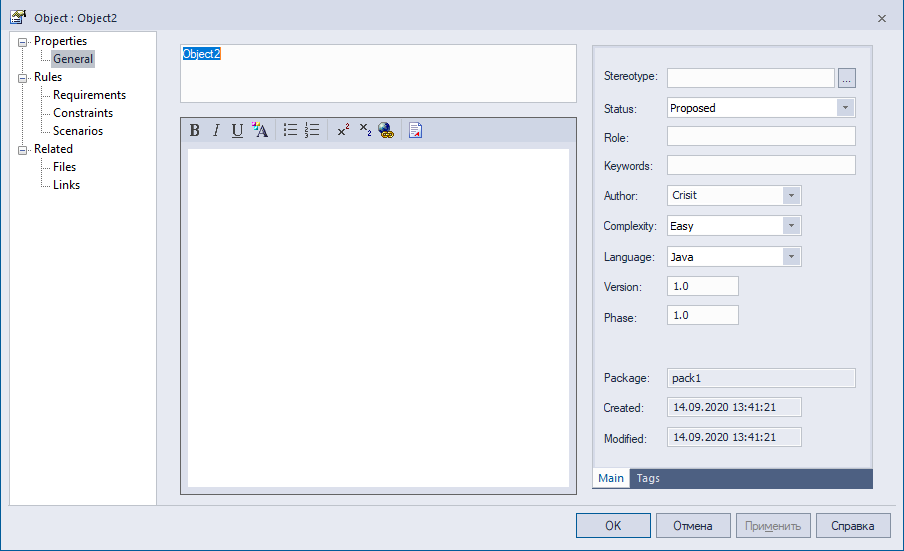


Figure 11 Creăm Object

* Pentru a crea un Object alegem din More tools secțiunea Object și adăugăm acest obiect.

**Concluzii:**

Enterprise Architect este o aplicație utilă și complexă care ne permite să creăm diverse diagrame care ne permite să elaborăm un algoritm precis pentru diferite aplicații și sisteme. Crearea unor asemenea diagrame ne ajută să proiectăm mai exact ale noastre lucrări. Putem să adăugăm diferite elemente din cadrul acestor proiecte ca actori, use case, putem să stabilim diferite relații dintre elemente, putem să stabilim care acțiuni sunt necesare dar care nu, etc.

Funcționalitatea acestei aplicații este impresionantă și eu consider că pe viitor voi putea proiecta niște sisteme complexe și detaliate.

**Bibliografie**

1. **Melnic R., Sava N.** Indrumar metodic “Analiza si modelarea sistemelor informationale”.
2. **Моделирование бизнес процессов|CASE средства|Rational Rose**, [Электронный ресурс].-Режим доступа: <http://www.kpms.ru/Automatization/Rational_Rose.htm>
3. Doug Rosenberg. "Service Oriented Architecture Roadmap". Iconix.
4. https://sparxsystems.com/resources/user-guides/15.2/index.html#fundamentals