



Università degli Studi dell'Aquila



Dipartimento di Ingegneria e Scienze
dell'Informazione e Matematica

Corso di Laurea in Informatica

Generation of Textual Modelling Environments for Metamodel-specific Languages

Relatore:
Prof.
Alfonso Pierantonio

Candidato:
Stefano Valentini
Matr.: 227718

Introduzione

- Questa tesi descrive un processo automatico per la *generazione* di ambienti di modellazione testuali Web-based. In particolare, partendo da un meta-modello, è stato definito:
 - *un mapping canonico per la definizione della sintassi testuale (concreta);*
 - *la generazione del relativo ambiente di modellazione.*

L'intero processo è realizzato da un applicativo Java a linea di comando che utilizza l'interazione di motori di trasformazione di modelli e frameworks quali *Acceleo* e *Xtext*.



Model Driven Engineering

- Si è fatto uso di tecnologie proprie della **Model Driven Engineering**. **MDE** è una metodologia di sviluppo software che si basa su concetti chiave quali:
 - **Modelli**: dato uno scopo, possiamo definire un *modello* come un artefatto che rappresenta un certo sistema astraendone i dettagli non utili all'ottenimento di tale scopo.
 - **Meta-modelli**: un *meta-modello* è una specifica formale per la creazione di modelli.
 - **Trasformazioni di modelli**: programmi per la generazione automatica di nuovi modelli a partire da modelli e/o artefatti testuali già esistenti.
- MDE fa riferimento ad una **gerarchia di meta-modellazione** per la quale:
 - nel livello inferiore, **M0**, si trova il sistema da modellare;
 - nel livello **M1** si trova il modello (o i modelli) che rappresentano tale sistema;
 - per ogni modello, nel livello **M2**, si trova il rispettivo meta-modello a cui esso è conforme;
 - analogamente, nel livello **M3**, per ogni meta-modello, si trova il meta meta-modello a cui è conforme.

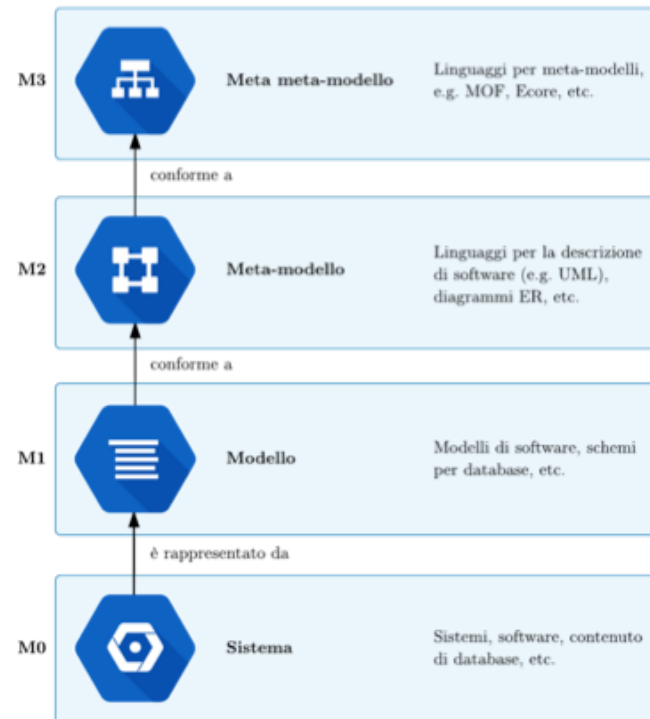


Fig. 1. Gerarchia di Meta-modellazione.

Eclipse Modeling Framework

- Gli strumenti utilizzati fanno parte dell'**Eclipse Modeling Framework (EMF)** è un framework che rende disponibili tools per la manipolazione di modelli. In particolare EMF fornisce un meta-modello per la creazione di modelli chiamato **Ecore**. I concetti principali dello standard Ecore sono le *EClass*, *EStructuralFeature*, *EReference* ed *EAttribute*. In generale, una EClass può contenere più EStructuralFeature ovvero EReference e/o EAttribute.

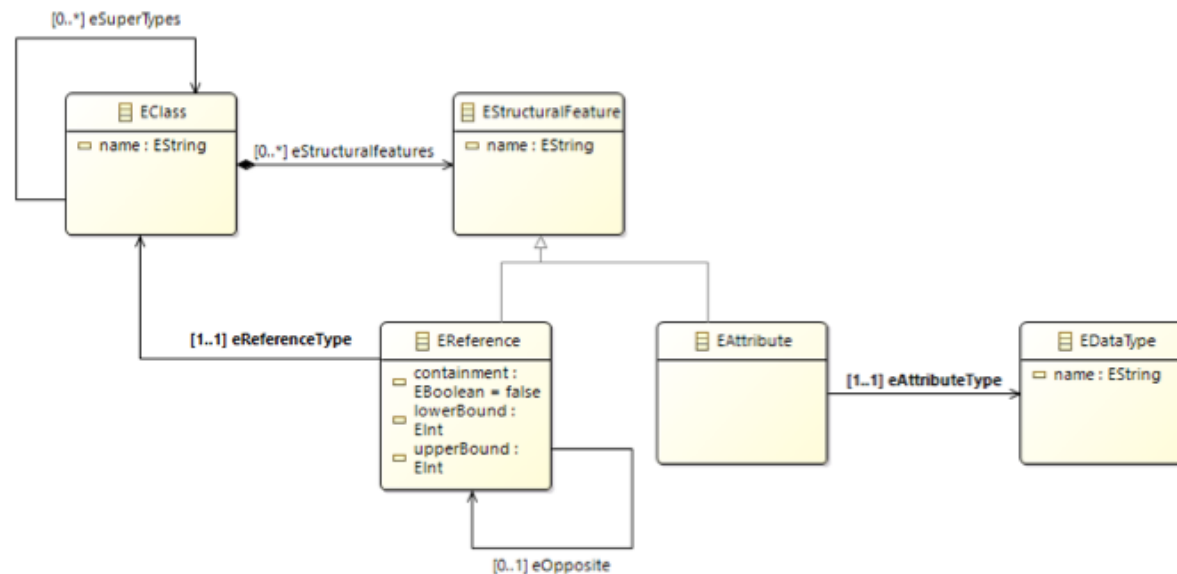


Fig. 2. Struttura dello standard Ecore.

Acceleo

- Acceleo è un linguaggio di trasformazioni M2T. Una trasformazione M2T in Acceleo è composta da uno o più **moduli**, ovvero file *.mtl* contenenti **tempaltes** e **queries**.
 - Le **templates** sono porzioni di codice delimitate dai tag `[template...][/template]`, contenenti istruzioni per la generazione di testo.
 - Le **queries** sono porzioni di codice delimitate dal tag `[query... /]` e vengono utilizzate per interrogare e quindi estrarre informazioni dai modelli in input.
-
- Esempio di template Acceleo che genera un JavaBean per ogni classe contenuta in un modello UML dato in input.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://www.eclipse.org/uml2/3.0.0/UML' )]
3
4 [template public generate(aClass : Class)]
5 [file (aClass.name.concat('.java'), false)]
6     public class [aClass.name.toUpperFirst()] {
7         [for (p: Property | aClass.attribute) separator('\n')]
8             private [p.type.name/] [p.name/];
9         [/for]
10
11         [for (p: Property | aClass.attribute) separator('\n')]
12             public [p.type.name/] get[p.name.toUpperFirst()]/() {
13                 return this.[p.name/];
14             }
15         [/for]
16
17         [for (o: Operation | aClass.ownedOperation) separator('\n')]
18             public [o.type.name/] [o.name/]() {
19                 // TODO should be implemented
20             }
21         [/for]
22     }
23 [/file]
24 [/template]
```

Fig. 3. Esempio di un file *.mtl*.

Xtext

- Xtext è un framework di Eclipse per l'implementazione di linguaggi di programmazione e DSLs a partire da una **specifica grammaticale**. Da questa specifica Xtext genera un meta-modello Ecore rappresentante le entità espresse nella grammatica ed il parser associato; Xtext offre inoltre la possibilità di integrare il progetto con il supporto per Web editor testuali. Gli editor sono implementati in JavaScript. Le risorse specifiche del linguaggio e servizi di assistenza (quali **check sintattici**, **syntax highlighting**, e **code completion**) vengono forniti tramite richieste HTTP alla componente server (anch'essa generata da Xtext).
- La **specifica grammaticale** descrive la sintassi concreta e come è rappresentata in memoria. È composta da diversi tipi di regole:
 - **Regole terminali**: descrivono i token del DSL, solitamente sono utilizzate per esprimere tipi di dato base quali INT, STRING etc.
 - **Regole di parsing**: descrivono le entità del DSL, sono utilizzate come pattern per la produzione di EClass nel modello Ecore derivato dalla grammatica.
 - **Regole di tipo**: sono utilizzate per esprimere tipi di dato complesso, creano istanze di EDataType invece che di Eclass,
 - **Regole di enumerazione**: possono essere viste come scorciatoie per le regole di tipo; permettono di definire un insieme di possibili "opzioni" per una data regola.

```
1 import "http://www.eclipse.org/emf/2002/Ecore" as.ecore
2 ...
3 terminal INT returns.ecore::EInt:
4     ('0'..'9')+;
```

Fig. 4.1. Esempio di regola terminale.

```
1 Entity:
2     'entity' name = ID ('extends' superType=[Entity])? '{'
3         attributes += Attribute*
4         '}';
```

Fig. 4.2. Esempio di regola di parsing.

```
1 QualifiedName returns.ecore::EString:
2     ID ('.' ID)*;
```

Fig. 4.3. Esempio di regola di tipo.

```
1 enum METHOD returns.METHOD
2     GET = 'GET' | POST = 'POST' ;
```

Fig. 4.4. Esempio di regola di enumerazione.

Implementazione

- L'applicativo sviluppato esegue i seguenti passi: dopo aver *caricato e registrato* il meta-modello Ecore, e dopo aver *compilato* la template Acceleo, esegue la *trasformazione M2T* che genera il file *.xtext* contenente la *specifica grammaticale* prodotta dal meta-modello in input. Questo file è inserito in un apposito *progetto Xtext* costruito dall'applicativo stesso; la *compilazione* di questo progetto porta alla generazione dell'editor Web e della componente server che si occupa di fornire i servizi di assistenza.

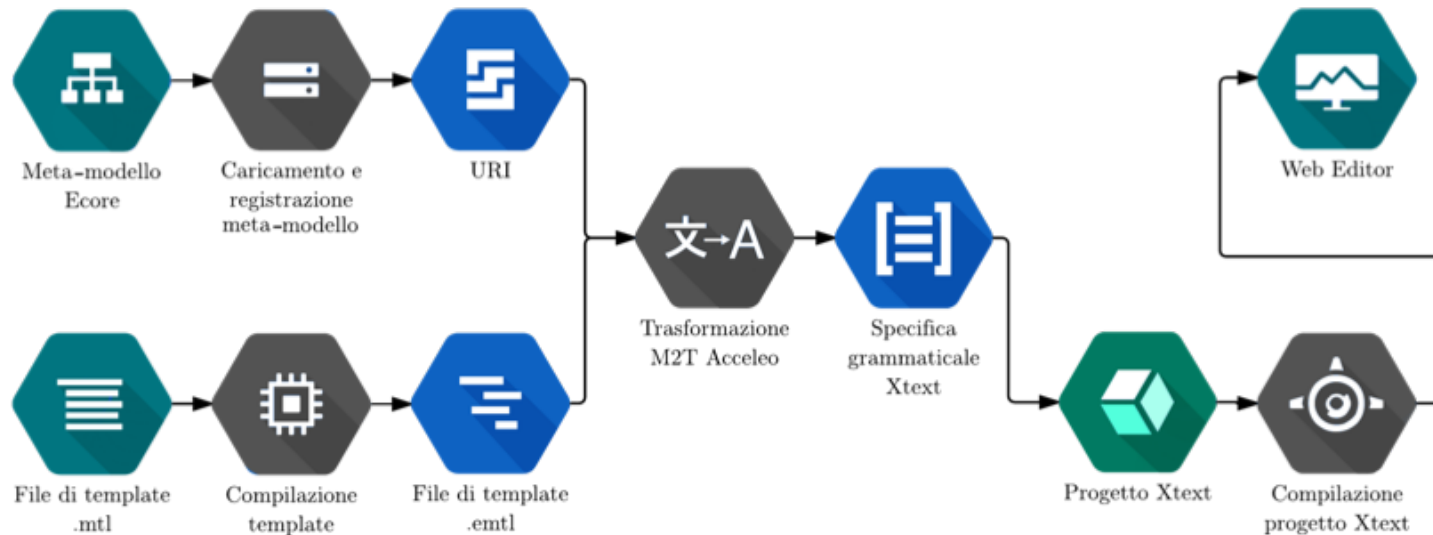


Fig. 5. Flusso di esecuzione dell'applicativo.

Template Acceleo *generate.mtl*

- L'essenza dell'applicativo risiede nella template; per ottenere un file *.xtext*, è necessario fornire alla template Acceleo l'URI del meta-modello Ecore in input e l'entry point. L'**entry point** è il nome della EClass da dove parte la generazione, è da intendere come la principale entità del meta-modello o l'entità che in qualche modo "racchiude" le altre entità. La template Acceleo genera le regole grammaticali in questo modo:
 - viene analizzata la EClass **entry point**, generando *regole di parsing* che riflettono le sue caratteristiche (attributi, relazioni, composizioni; considerando il caso in cui possa essere superclasse di altre EClass e/o una EClass interfaccia/astratta).
 - con lo stesso procedimento vengono generate le *regole di parsing* per le altre **EClass**.
 - viene generata una *regola di tipo* per ogni **EDataType** nel meta-modello in input.
 - vengono generate *regole di enumerazione* per ogni **EEnum** nel meta-modello in input.

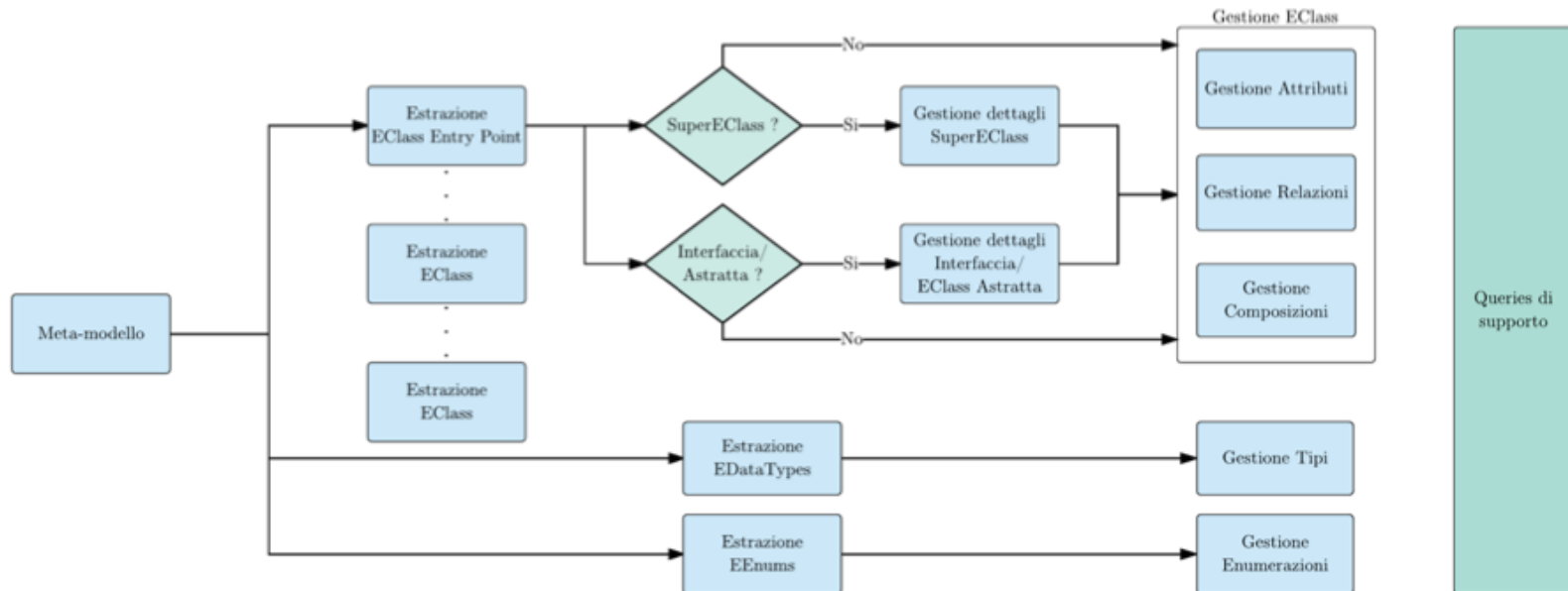


Fig. 6. Struttura della template Acceleo *generate.mtl*

Esempio di funzionamento

- Prendiamo in esame il meta-modello *school.ecore* in Fig. 7. Da questo meta-modello, attraverso la trasformazione M2T Acceleo otteniamo la specifica grammaticale in Fig. 8. Questo file viene successivamente posizionato all'interno del progetto Xtext appositamente generato. Dopo aver compilato il progetto, otteniamo l'editor Web in Fig. 9.

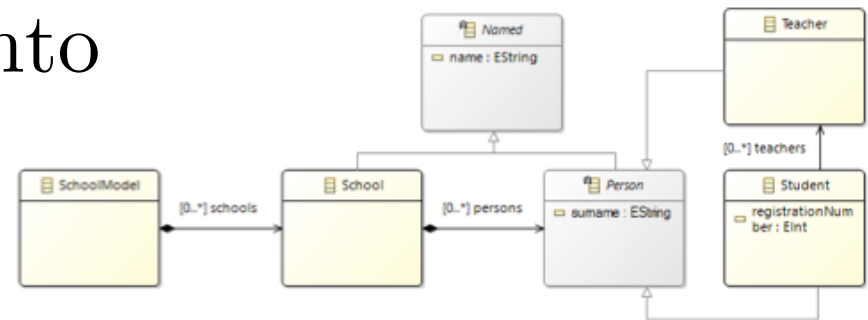


Fig. 7. Meta-modello *school.ecore*.

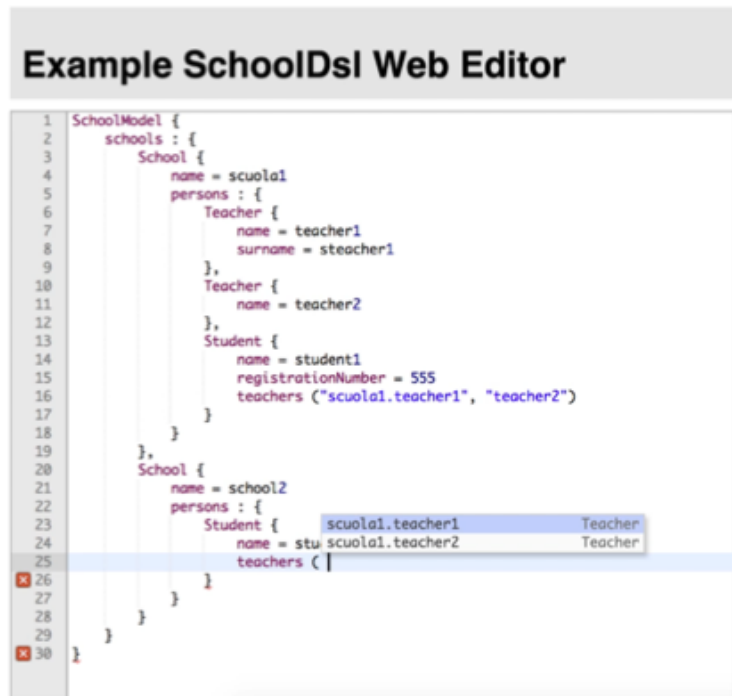


Fig. 9. Editor Web per il meta-modello *school.ecore*.

```

1  grammar org.xtext.schoolDsl.SchoolDsl
2      with org.eclipse.xtext.common.Terminals
3  generate schoolcoreDsl "http://www.xtext.org/schoolcoreDsl"
4  import "http://www.eclipse.org/emf/2002/Ecore" as.ecore
5
6  SchoolModel returns SchoolModel:
7      {SchoolModel}
8      'SchoolModel'
9      '{'
10         ('schools' ':' '{' schools+=School ( "," schools+=School)* '}' )?
11         '}' ;
12
13  Named returns Named:
14      School | Student | Teacher ;
15  Person returns Person:
16      Student | Teacher ;
17
18  School returns School:
19      {School}
20      'School'
21      '{'
22         ('name' '=' name = EString )?
23         ('persons' ':' '{' persons+=Person ( "," persons+=Person)* '}' )?
24         '}' ;
25
26  Student returns Student:
27      {Student}
28      'Student'
29      '{'
30         ('name' '=' name = EString)?
31         ('surname' '=' surname = EString)?
32         ('registrationNumber' '=' registrationNumber = EInt)?
33         ('teachers' '{' teachers += [Teacher|STRING]
34             ( "," teachers += [Teacher|STRING])* '}' )?
35         '}' ;
36
37  Teacher returns Teacher:
38      {Teacher}
39      'Teacher'
40      '{'
41         ('name' '=' name = EString)?
42         ('surname' '=' surname = EString)?
43         '}' ;
44  EInt returns.ecore::EInt:
45      '-' ? INT;
46  EString returns.ecore::EString:
47      STRING | ID;

```

Fig. 8. Grammatica generata dal meta-modello *school.ecore*.

Caso di studio: generazione dell'ambiente di modellazione Web per il meta meta-modello *ecore.ecore*

- Avendo un generatore di ambienti di modellazione possiamo sperimentare con meta-modelli più complessi; in particolare, essendo il **meta meta-modello per lo standard Ecore** comunque considerato un meta-modello, possiamo generare, utilizzando l'applicativo sviluppato, un ambiente di sviluppo testuale per i meta-modelli Ecore stessi. Considerando come entry point l'EClassifier *EPackage*, quello che otteniamo è una specifica grammaticale che rispecchia i vincoli di modellazione espressi dallo standard stesso; ovvero otteniamo in output un ambiente di modellazione Web-based per generici meta-modelli conformi allo standard Ecore. In Fig. 10 è mostrato un esempio pratico di modellazione; nel dettaglio, tramite il Web editor generato, è stato costruito il meta-modello *school.ecore*, mostrato in Fig. 7 (precedente slide).



```
1 EPackage {
2   eClassifiers : {
3     EDataType {
4       name = EInt
5     },
6     EDataType {
7       name = EString
8     },
9     EClass {
10      name = Named
11      abstract = true
12      eStructuralFeatures : {
13        EAttribute {
14          name = "name" //name w/out ' ' is a keyword
15          eType ("EString")
16        }
17      }
18    },
19    EClass {
20      name = Person
21      abstract = true
22      eSuperTypes ("Named")
23      eStructuralFeatures : {
24        EAttribute {
25          name = Named
26          eType Person
27        }
28      }
29    },
30    EClass {
31      name = School
32      eSuperTypes ()
33      eStructuralFeatures : {
34        EReference {
```



```
35      name = persons
36      upperBound = -1
37      containment = true
38      eType ("Person")
39    }
40  },
41 },
42 EClass {
43   name = SchoolModel
44   eStructuralFeatures : {
45     EReference {
46       name = schools
47       upperBound = -1
48       containment = true
49       eType ("School")
50     }
51   }
52 },
53 EClass {
54   name = Teacher
55   eSuperTypes ("Person")
56 },
57 EClass {
58   name = Student
59   eSuperTypes ("Person")
60   eStructuralFeatures : {
61     EAttribute {
62       name = registrationNumber
63       eType ("EInt")
64     },
65     EReference {
66       name = teachers
67       upperBound = -1
68       eType ("Teacher")
69     }
70   }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }
```

Fig. 10. Editor Web per il meta meta-modello *ecore.ecore*