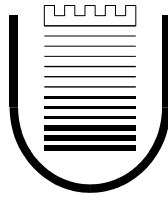


Università degli Studi di Roma "Tor Vergata"



Facoltà di Ingegneria

**Tesi di Laurea Magistrale in
Ingegneria Informatica**

Qualificazione di nuove tecnologie

*Da un approccio innovativo, basato su
modellazione degli obiettivi, opinioni di esperti e
metodo Monte Carlo, a uno strumento software
a supporto*

Relatore

Prof. Ing. Giovanni Cantone

Candidato

Stefano Di Alesio

Correlatore

Dr. Ing. Davide Falessi

Anno Accademico 2009/2010

FAV_QUOTE_PLACEHOLDER

Indice

Indice	3
Figure.....	10
Tabelle	14
Abstract.....	15
Contesto e Motivazioni	15
Condizioni iniziali e Obiettivi	15
Risultati	16
Conclusioni	18
1 Contesto	19
1.1 Sistemi tecnologici e safety-critical	19
1.1.1 Dependability	20
1.1.1.1 Attributi.....	22
1.1.1.2 Minacce.....	23
1.1.1.3 Mezzi.....	24
1.1.2 Rischio	25
1.1.2.1 Gestione del rischio	29
1.2 Qualificazione di tecnologie.....	31
1.2.1 Il processo di qualificazione secondo DNV	33
1.3 Motivazioni.....	35
1.3.1 Scarsa organizzazione delle informazioni necessarie per la valutazione	36
1.3.2 Tardiva raccolta delle evidenze	37
1.3.3 Modalità e meccanismi di fallimento non specificati	37
1.3.4 Inadeguato supporto decisionale per il miglioramento del sistema	38
1.3.5 Inadeguato supporto per l'identificazione dell'impatto dei cambiamenti.....	38
1.3.6 Trattamento implicito dell'ambiente operativo	39

2	<i>Modus: un approccio innovativo per la qualificazione di nuove tecnologie ...</i>	40
2.1	Visione d'insieme	40
2.2	Specifica degli obiettivi strategici della tecnologia.....	44
2.3	Decomposizione degli obiettivi.....	45
2.3.1	Decomposizione degli obiettivi secondo la notazione KAOS	46
2.3.2	Dimensioni della decomposizione degli obiettivi.....	48
2.3.3	Euristiche di modellazione degli obiettivi.....	50
2.4	Identificazione degli ostacoli	51
2.5	Pianificazione della raccolta delle evidenze.....	53
2.6	Elicitazione delle quantità dagli esperti.....	56
2.6.1	Descrizione delle quantità che gli esperti devono specificare	57
2.6.2	Processo di elicitazione	58
2.6.3	Elicitazione di distribuzioni sconosciute.....	61
2.6.4	Gestione degli errori in fase di elicitazione	63
2.7	Analisi del soddisfacimento degli obiettivi strategici	65
2.7.1	Propagazione dei valori puntuali di probabilità	65
2.7.2	Propagazione delle distribuzioni di probabilità.....	69
2.8	Miglioramento di obiettivi, evidenze e stime degli esperti.....	73
2.8.1	Analisi di sensibilità	73
2.8.2	Analisi dell'impatto dei cambiamenti.....	74
3	<i>Modus Tool: l'applicazione ideata e sviluppata</i>	80
3.1	Obiettivi	80
3.2	Visione d'insieme.....	81
3.2.1	Glossario.....	81
3.2.2	Diagramma generale di attori e casi d'uso	84
3.2.3	Activity diagram generale.....	86
3.2.4	Pattern architetturale BCE	89
3.2.5	Enterprise Architect come piattaforma di riferimento	91
3.3	Caso d'uso Goal Modeling: dalla definizione alla sua implementazione	92
3.3.1	Definizione.....	92
3.3.1.1	Decomposizione di goal o ostacoli per riferimento	94

3.3.2	Activity diagrams: specifica delle operazioni	104
3.3.2.1	Creazione Goal Model.....	105
3.3.2.2	Visualizzazione Goal Model	107
3.3.3	Progettazione: MDG Technologies e profili UML	108
3.3.4	Screenshots	109
3.3.4.1	Creazione di un Goal Model.....	109
3.3.4.2	Creazione di un Goal Diagram	110
3.3.4.3	Creazione di un elemento.....	111
3.3.4.4	Creazione di una relazione standard	113
3.3.4.5	Creazione di una relazione di decomposizione per riferimento	114
3.4	Caso d'uso Model Validation: dalla definizione alla sua implementazione	115
3.4.1	Definizione.....	115
3.4.2	Activity diagrams: specifica delle operazioni	117
3.4.2.1	Avvio validazione Goal Model.....	117
3.4.2.2	Validazione Goal Model	119
3.4.2.2.1	Controllo oggetto	120
3.4.3	Progettazione: regole di consistenza e framework di EA	121
3.4.4	Screenshots	122
3.4.4.1	Modifica file delle regole	122
3.4.4.2	Selezione di un goal model ed avvio della validazione	124
3.4.4.3	Visualizzazione del report di validazione	125
3.5	Caso d'uso Elicitation: dalla definizione alla sua implementazione.....	126
3.5.1	Definizione.....	126
3.5.2	Activity diagrams: specifica delle operazioni	127
3.5.2.1	Creazione sessione di elicitazione.....	128
3.5.2.2	Memorizzazione sessione di elicitazione	129
3.5.2.2.1	Scansione goal diagram.....	130
3.5.2.3	Visualizzazione sessione di elicitazione	132
3.5.2.3.1	Visualizzazione dettaglio sessione	134
3.5.2.4	Compilazione questionari	136
3.5.2.5	Salvataggio sessione di elicitazione	138
3.5.2.6	Export sessione di elicitazione	140
3.5.3	Progettazione: analisi del problema e scelte architetture	142
3.5.4	Screenshots	145
3.5.4.1	Avvio della feature e finestra principale	145
3.5.4.2	Creazione di una sessione.....	148

3.5.4.3	Selezione di una sessione	150
3.5.4.4	Visualizzazione di una snapshot di un diagramma	150
3.5.4.5	Visualizzazione del dettaglio di un'entità del diagramma	152
3.5.4.6	Elicitazione di un'opinione.....	154
3.5.4.7	Salvataggio di una sessione	155
3.5.4.8	Export di una sessione	157
4	<i>Conclusioni e sviluppi futuri</i>	159
4.1	Sviluppi futuri	161
	<i>Bibliografia.....</i>	163
A	<i>Appendice A: Specifica dei requisiti utente di Modus Tool</i>	166
A.1	Goal Modeling	166
A.2	Model Validation	168
A.3	Elicitation.....	173
B	<i>Appendice B: Specifica dei casi d'uso di Modus Tool</i>	176
B.1	Goal Modeling	176
B.2	Model Validation	184
B.3	Elicitation.....	189
C	<i>Appendice C: Specifica dell'architettura di Modus Tool</i>	206
C.1	Goal Modeling	206
C.1.1	Modus Goal Model Template.....	206
C.1.2	Modus Goal Diagram Profile	208
C.1.3	Modus Entities Profile	209
C.1.4	Shape Scripts	213
C.1.5	Modus Toolbox Profile	214
C.2	Model Validation	217
C.2.1	File delle regole di consistenza.....	217
C.2.2	Architettura: overview e sottosistemi.....	219
C.2.2.1	Sottosistema Rules.....	219
C.2.2.2	Sottosistema Output.....	221
C.3	Elicitation.....	223

C.3.1	Architettura: overview e sottosistemi.....	223
C.3.1.1	Package Boundary.....	224
C.3.1.2	Package Control	226
C.3.1.3	Package Entity.....	229
C.3.1.4	Sottosistema Data.....	232
C.3.1.5	Sottosistema Serialization.....	236

D **Appendice D: Codice sorgente di Modus Tool.....239**

D.1 Main Package.....239

D.1.1	Main.cs	239
	private static EA.Repository _repository;.....	239
D.1.2	Configuration.cs	242

D.2 Boundary Package.....243

D.2.1	ElicitProbabilitiesForm.cs	243
D.2.2	ElicitProbabilitiesForm.Designer.cs	257
D.2.3	ExportSessionForm.cs	268
D.2.4	ExportSessionForm.Designer.cs	272
D.2.5	NewSessionForm.cs.....	276
D.2.6	NewSessionForm.Designer.cs	281
D.2.7	QuestionnaireForm.cs	284
D.2.8	QuestionnaireForm.Designer.cs	287
D.2.9	SaveSessionForm.cs.....	290
D.2.10	SaveSessionForm.Designer.cs.....	294
D.2.11	SnapshotZoomForm.cs	297
D.2.12	SnapshotZoomForm.Designer.cs	298

D.3 Control Package299

D.3.1	BlackboardChangesControl.cs	299
D.3.2	DataPackageControl.cs	300
D.3.3	DeleteSessionControl.cs	301
D.3.4	ElicitationChangesControl.cs	301
D.3.5	ExportSessionControl.cs	303
D.3.6	NewSessionControl.cs	316
D.3.7	OutputControl.cs.....	318
D.3.8	QuestionnaireChangesControl.cs	319
D.3.9	RulesControl.cs	321
D.3.10	SaveSessionControl.cs	322

D.3.11	SessionChangesControl.cs	324
D.3.12	ShowSessionsControl.cs	325
D.4	Data Package	332
D.4.1	DataDecomposition.cs	332
D.4.2	DataDiagram.cs	336
D.4.3	DataDomainEntity.cs	339
D.4.4	DataElement.cs	341
D.4.5	DataElicitation.cs	344
D.4.6	DataExpert.cs	350
D.4.7	DataSession.cs	352
D.4.8	PackageManager.cs	360
D.5	Entity Package.....	364
D.5.1	Decomposition.cs	364
D.5.2	Diagram.cs	365
D.5.3	DomainEntity.cs	368
D.5.4	Element.cs	369
D.5.5	Elicitation.cs	371
D.5.6	ElicitedEntity.cs	372
D.5.7	Expert.cs	374
D.5.8	ModusStereotype.cs.....	376
D.5.9	ModusTaggedValue.cs.....	378
D.5.10	Probability.cs	379
D.5.11	Question.cs	380
D.5.12	Questionnaire.cs.....	380
D.5.13	Session.cs.....	383
D.5.14	SnapShot.cs	387
D.5.15	StandardQuestionnaire.cs	389
D.6	Output Package.....	393
D.6.1	OutputManager.cs	393
D.7	Resources Package	395
D.7.1	Data/MDG	395
D.7.1.1	ModusAddinTechnology.mts	395
D.7.1.2	ModusAddinTechnology.xml	396
D.7.1.3	ModusDiagramProfile.xml	409
D.7.1.4	ModusElementsProfile.xml.....	410

D.7.1.5	ModusGoalModelTemplate.xml	413
D.7.1.6	ModusToolboxProfile.xml	415
D.7.2	Data/Reference Data	422
D.7.2.1	ModusProjectRoles.xml	422
D.7.2.2	ModusTaggedValues.xml	422
D.7.3	Data/Rules	423
D.7.3.1	Rules.csv	423
D.7.4	ShapeScripts	426
D.7.4.1	Assumption.txt	426
D.7.4.2	Complete AND.txt	426
D.7.4.3	Complete OR.txt	427
D.7.4.4	Evidence.txt	427
D.7.4.5	Goal.txt	427
D.7.4.6	Obstacle.txt	427
D.7.4.7	Obstructs.txt	428
D.7.4.8	Outgoing AND link.txt	428
D.7.4.9	Partial AND.txt	429
D.7.4.10	Partial OR.txt	429
D.8	Rules Package	429
D.8.1	Rule.cs	429
D.8.2	RulesConfiguration.cs	432
D.8.3	RulesManager.cs	433
D.8.4	ValidationModel.cs	484
D.8.5	ValidationStatistics.cs	492
D.9	Serialization Package	494
D.9.1	SerializationObject.cs	494
D.9.2	SerializationQuestion.cs	494
D.9.3	SerializationQuestionnaire.cs	496
D.10	Utils Package	498
D.10.1	ExtensionMethods.cs	498
D.10.2	Tuple.cs	500

Figure

Figura 1: Tassonomia della dependability	22
Figura 2: Relazioni tra avarie, errori e difetti	24
Figura 3: Relazione tra i concetti legati alla sicurezza	26
Figura 4: Spazio del rischio	27
Figura 5: Concetti generali della riduzione del rischio	28
Figura 6: Matrice del livello di rischio	29
Figura 7: Interazioni fra il processo di sviluppo ed il processo di gestione dei rischi	30
Figura 8: Il processo di qualificazione secondo DNV	33
Figura 9: Visione d'insieme della metodologia Modus	41
Figura 10: Decomposizione degli obiettivi in KAOS.....	47
Figura 11: Dimensioni della decomposizione degli obiettivi	49
Figura 12: Diagramma di un ostacolo.....	51
Figura 13: Supporto degli obiettivi foglia attraverso evidenze	54
Figura 14: Classificazione delle evidenze	55
Figura 15: Funzione di densità della distribuzione triangolare	62
Figura 16: Funzione di distribuzione cumulativa della distribuzione triangolare	62
Figura 17: Esempio di distribuzione triangolare con gli intervalli di confidenza estesi.....	63
Figura 18: Regole per la propagazione dei valori puntuali di probabilità	66
Figura 19: Propagazione dei valori di probabilità nel modello in termini di variabili di input e variabili di output	69
Figura 20: Algoritmo Monte Carlo di propagazione delle distribuzioni di probabilità	70
Figura 21: Esempio di un modello degli obiettivi con le distribuzioni di probabilità degli obiettivi foglia	71
Figura 22: Esempio di distribuzione di probabilità risultante dall'esecuzione di una simulazione Monte Carlo per un obiettivo strategico	72
Figura 23: Risultati dell'analisi di sensibilità per l'obiettivo radice della Figura 21	74
Figura 24: Tabella riepilogativa dell'impatto di un cambiamento.....	75
Figura 25: Grafici di confronto dell'impatto di un cambiamento.....	76
Figura 26: Box plot relativo all'impatto di un cambiamento.....	77
Figura 27: Analisi dell'impatto relativa all'evoluzione di un input	79

Figura 28: Diagramma degli attori di Modus Tool.....	85
Figura 29: Diagramma dei casi d'uso principali di Modus Tool	86
Figura 30: Activity diagram generale dell'attività di qualificazione di nuove tecnologie secondo la metodologia Modus	88
Figura 31: Overview architetturale di Modus Tool	90
Figura 32: Disposizioni equivalenti degli elementi in un class diagram	95
Figura 33: Disposizioni equivalenti degli elementi in un goal diagram	96
Figura 34: Disposizione errata degli elementi in un goal diagram	97
Figura 35: Gradi di libertà di un class diagram	98
Figura 36: Gradi di libertà di un goal diagram.....	99
Figura 37: Goal tree decomposto in un singolo goal diagram	100
Figura 38: Goal tree decomposto in più goal diagram (diagramma principale).....	101
Figura 39: Goal tree decomposto in più goal diagram (diagramma secondario)	101
Figura 40: Gradi di libertà di un goal diagram con decomposizioni per riferimento	102
Figura 41: Struttura di decomposizione multilayer di un goal tree.....	103
Figura 42: Albero dei diagrammi di decomposizione di Goal1.....	104
Figura 43: Activity diagram della feature di Goal Modeling.....	105
Figura 44: Structured activity "Creazione Goal Model"	106
Figura 45: Structured activity "Visualizzazione Goal Model"	108
Figura 46: Creazione di un goal model	110
Figura 47: Creazione di un goal diagram	111
Figura 48: Creazione di un elemento	112
Figura 49: Visualizzazione dell'elemento creato nel project browser e nel diagramma.....	113
Figura 50: Creazione di una relazione standard	114
Figura 51: Creazione di una relazione di decomposizione per riferimento	114
Figura 52: Visualizzazione di una relazione di decomposizione per riferimento	115
Figura 53: Activity diagram della feature di Model Validation.....	117
Figura 54: Structured activity "Avvia validazione Goal Model"	118
Figura 55: Structured activity "Validazione Goal Model"	119
Figura 56: Structured activity "Controllo oggetto"	120
Figura 57: File Rules. csv contenente le definizioni delle regole di consistenza	123
Figura 58: Avvio della validazione	124
Figura 59: Status della validazione	124
Figura 60: Report della validazione nella tab "Model Validation" di EA	125
Figura 61: Report della validazione nella tab "Modus" di Modus Tool.....	125
Figura 62: Activity diagram della feature di Elicitation	128
Figura 63: Structured activity "Creazione sessione di elicitazione"	129

Figura 64: Structured activity "Memorizzazione sessione di elicitazione"	130
Figura 65: Structured activity "Scansione goal diagram"	131
Figura 66: Structured activity "Visualizzazione sessione di elicitazione"	133
Figura 67: Structured activity "Visualizzazione dettaglio sessione"	135
Figura 68: Structured activity "Compilazione questionari"	137
Figura 69: Structured activity "Salvataggio sessione di elicitazione"	139
Figura 70: Structured activity "Export sessione di elicitazione"	141
Figura 71: Modello dei dati della feature di Elicitation	143
Figura 72: Avvio della feature di Elicitation dell'add-in.....	146
Figura 73: Finestra principale della feature di Elicitation.....	147
Figura 74: Accesso utente e lista delle sessioni.....	148
Figura 75: Finestra di creazione di una sessione (1).....	149
Figura 76: Finestra di creazione di una sessione (2).....	149
Figura 77: Entità di una sessione e lavagna.....	150
Figura 78: Snapshot di un diagramma della sessione	151
Figura 79: Finestra di zoom di una snapshot.....	152
Figura 80: Informazioni sull'entità selezionata (1)	153
Figura 81: Informazioni sull'entità selezionata (2)	154
Figura 82: Finestra di elicitazione (1)	155
Figura 83: Finestra di elicitazione (2)	155
Figura 84: Finestra di salvataggio di una sessione	156
Figura 85: Avviso di modifiche non salvate	156
Figura 86: Finestra di export di una sessione (1).....	157
Figura 87: Finestra di export di una sessione (2).....	158
Figura 88: Diagramma dei requisiti della feature di Goal Modeling	166
Figura 89: Diagramma dei requisiti della feature di Model Validation	168
Figura 90: Diagramma dei requisiti della feature di Elicitation	173
Figura 91: Use case diagram della feature di Goal Modeling.....	176
Figura 92: Use case diagram della feature di Model Validation	184
Figura 93: Use case diagram della feature di Elicitation	190
Figura 94: Profilo di definizione del tipo di diagramma "Goal Diagram"	209
Figura 95: Profilo di definizione delle entità di Modus Tool (stereotipi di Class).....	210
Figura 96: Profilo di definizione delle entità di Modus Tool (stereotipi di Association con l'attributo directed)	210
Figura 97: Profilo di definizione delle entità di Modus Tool (stereotipi di Association)	210
Figura 98: Profilo di definizione delle entità di Modus Tool (stereotipi di n-ary Association)	211
Figura 99: Profilo di definizione delle entità di Modus Tool (stereotipi di Artifact).....	211

Figura 100: Profilo di definizione della toolbox di Modus Tool (elementi della toolbox)	215
Figura 101: Profilo di definizione della toolbox di Modus Tool (immagini della toolbox, 1).....	215
Figura 102: Profilo di definizione della toolbox di Modus Tool (immagini della toolbox, 2).....	216
Figura 103: Profilo di definizione della toolbox di Modus Tool (immagini della toolbox, 3).....	216
Figura 104: Package diagram della feature di Model Validation	219
Figura 105: Class diagram del sottosistema "Rules"	220
Figura 106: Class diagram del sottosistema "Output"	222
Figura 107: Package diagram della feature di Elicitation	224
Figura 108: Class diagram del package "Control" della feature di Elicitation (1)	227
Figura 109: Class diagram del package "Control" della feature di Elicitation (2)	227
Figura 110: Class diagram del package "Entitiy" della feature di Elicitation	230
Figura 111: Class diagram del sottosistema "Data" della feature di Elicitation	234
Figura 112: Class diagram del sottosistema "Serialization" della feature di Elicitation	237

Tabelle

Tabella 1: Parole chiave per la definizione degli obiettivi	45
Tabella 2: Visione d'insieme del processo di elicitazione	59
Tabella 3: Algoritmo di propagazione dei valori puntuali	68
Tabella 4: Glossario dei termini.....	84
Tabella 5: Struttura gerarchica di un goal model.....	93
Tabella 6: Metaclassi e metassociazioni della feature di Goal Modeling.....	94
Tabella 7: File MTS della Modus MDG Technology	109
Tabella 8: Requisiti utente della feature di Goal Modeling.....	167
Tabella 9: Requisiti utente della feature di Model Validation	169
Tabella 10: Regole di consistenza.....	172
Tabella 11: Requisiti utente della feature di Elicitation	175
Tabella 12: Use case report della feature di Goal Modeling	184
Tabella 13: Use case report della feature di Model Validation	189
Tabella 14: Use case report della feature di Elicitation.....	205
Tabella 15: Definizione del template di Goal Modeling	208
Tabella 16: Tabella riepilogativa del profilo UML delle entità di Modus Tool	212
Tabella 17: Shape Script di un goal	214
Tabella 18: Attributi delle regole di consistenza	218
Tabella 19: Descrizione delle classi del sottosistema "Rules"	221
Tabella 20: Descrizione delle classi del sottosistema "Output"	223
Tabella 21: Descrizione delle classi del package "Boundary"	226
Tabella 22: Descrizione delle classi del package "Control" della feature di Elicitation	229
Tabella 23: Descrizione delle classi del package "Entity" della feature di Elicitation	231
Tabella 24: Struttura del root node Modus_Data	232
Tabella 25: Descrizione dei tagged values degli stereotipi di persistenza	234
Tabella 26: Descrizione delle classi del sottosistema "Data" della feature di Elicitation	236
Tabella 27: Descrizione delle classi del sottosistema "Serialization" della feature di Elicitation.....	238

Abstract

Contesto e Motivazioni

In un mondo in cui la tecnologia è sempre più indispensabile per lo svolgersi delle attività umane, ci si pone il problema della valutazione della sicurezza funzionale di sistemi di grandi dimensioni, i cui malfunzionamenti potrebbero causare conseguenze tragiche per l'ambiente operativo e per i loro utilizzatori. È quindi strettamente necessario assicurare che i sistemi *safety-critical* rispettino requisiti di sicurezza funzionale (*dependability*). Esistono infatti discipline, come l'analisi e la gestione del rischio, che attraverso lo studio di obiettivi strategici (*strategic goal*) di un sistema mirano ad assicurare un livello di *dependability* desiderato. Tuttavia, se per tecnologie già affermate esistono degli standard di sicurezza riconosciuti con cui è possibile ottenere una certificazione da enti di terze parti, per le nuove tecnologie non è possibile procedere in tal senso.

Condizioni iniziali e Obiettivi

Per tali motivi, nel presente lavoro di Tesi Specialistica in Ingegneria Informatica si è affrontato il tema della qualificazione delle nuove tecnologie nell'ambito della gestione del rischio e della valutazione della sicurezza funzionale in sistemi tecnologici e *safety-critical*. In particolare, una volta identificata e acquisita la metodologia *Modus* [1], avendo rilevato che la proposta della stessa è del tutto recente e che trattasi di una metodologia decisamente innovativa per la qualificazione di nuove tecnologie, sviluppata dal Simula Research Laboratory di Oslo [2] in collaborazione con Det Norske Veritas [3] nell'ambito del progetto di

ricerca ModelME! [4], abbiamo ideato e sviluppato *Modus Tool*, un plugin di Enterprise Architect [5] a supporto di Modus.

Modus è un approccio che combina la modellazione degli obiettivi strategici di un sistema (*goal modeling*), l'elicitazione del parere degli esperti in merito alla probabilità di soddisfacimento di tali obiettivi (*elicitation*) e l'analisi quantitativa dei dati forniti dagli esperti attraverso l'utilizzo di strumenti matematici come il metodo Monte Carlo (*probabilistic simulation*). Come accade in casi analoghi, un supporto informatico ad un processo attuato in sistemi di grandi dimensioni risulta, laddove non indispensabile, comunque di estrema utilità.

L'obiettivo principale dello studio consisteva pertanto nello sviluppo di uno strumento che unificasse i vari aspetti di Modus in un'implementazione coerente. L'applicazione avrebbe dovuto fornire un ambiente di modellazione per la specifica degli obiettivi, delle interfacce per il supporto all'elicitazione delle opinioni degli esperti e alla loro memorizzazione persistente, nonché un supporto alla fase di analisi quantitativa delle probabilità oggetto di studio. Vincoli progettuali limitavano la scelta dell'ambiente escutivo al framework offerto da Enterprise Architect [5] ed imponevano l'interoperabilità con il tool @RISK [6], plugin di Microsoft Excel [7] utilizzato per le analisi statistiche.

Risultati

Lo sviluppo dell'applicazione è avvenuto nel rispetto di modelli industriali di processo software, come RUP [8], coprendo le fasi di ideazione, analisi e costruzione. Particolare attenzione è stata posta alla gestione dei requisiti, dei casi d'uso ed alla specifica dell'architettura. Il tool è stato concepito per focalizzarsi sui primi due aspetti della metodologia (modellazione degli obiettivi e supporto all'elicitazione), interessando il terzo aspetto (analisi probabilistica) solo in merito alla condivisione di un formato dati interpretabile da @RISK.

La prima parte del lavoro è stata incentrata sullo studio della metodologia, che ha portato alla definizione di un profilo UML per il goal modeling, il cui utilizzo si configura comunque come un compromesso tra un approccio formale ed una soluzione ad-hoc atto a soddisfare i requisiti utente di Modus Tool. Data la natura fortemente strutturata dei goal model, in questa fase si è ideato un supporto per la rappresentazione di un modello articolata in più diagrammi complementari.

Successivamente, è stato definito un insieme formale di regole di consistenza, base della funzionalità di validazione strutturale di goal model offerta da Modus Tool. Il supporto fornito all'utente in tal senso è consistito nella generazione di un report al termine della validazione stessa in cui fossero riportati gli errori individuati in fase di analisi del modello ed i suggerimenti per la loro correzione. È bene precisare come le prime due fasi del lavoro abbiano contribuito ad un'ulteriore formalizzazione nella definizione dei goal model introdotta in Modus a partire dalla notazione KAOS [9].

La parte finale del lavoro si è invece basata sull'attività di elicitazione delle opinioni degli esperti, consistendo nella progettazione di un'applicazione a supporto di essa. Particolare attenzione si è posta nella separazione delle responsabilità delle varie figure professionali coinvolte nell'attività, nel controllo della consistenza dei dati, nella comunicazione tra gli utenti, nelle modalità di condivisione delle informazioni e nella presentazione delle stesse. A causa dell'iteratività del processo di elicitazione, si è definito un framework operativo basato su *sessioni* di lavoro, in modo da consentire il mantenimento di uno storico delle attività precedenti utile al monitoraggio di dati immessi in diversi periodi temporali e alla loro eventuale rivisitazione.

Conclusioni

Gli obiettivi dello studio, riguardanti lo sviluppo di un applicazione che supportasse l'utente durante lo svolgimento del processo di qualificazione definito da Modus, sono stati conseguiti con successo: lo strumento sviluppato si configura come un assetto completo ed intuitivo a supporto della modellazione degli obiettivi strategici in termini di sicurezza funzionale di un sistema, della validazione strutturale dei modelli creati dall'utente e dell'elicitazione delle opinioni degli esperti in merito ad eventi rappresentati dagli obiettivi modellati.

Modus Tool è stato utilizzato con successo in ambito industriale nel progetto pilota studiato durante la definizione di Modus, ed è attualmente utilizzato in diversi altri progetti pilota [1].

1 Contesto

FAV_QUOTE_PLACEHOLDER

1.1 Sistemi tecnologici e safety-critical

I moderni sistemi tecnologici sono spesso combinazione di dispositivi meccanici ed elettronici controllati da software *embedded*. Questa combinazione porta ad enormi miglioramenti nelle prestazioni e nella flessibilità di un prodotto, trasformando a tutti gli effetti tradizionali dispositivi meccanici ed elettronici in *smart device* [10]. Lettori DVD, pacemaker ed automobili di nuova generazione sono tutti esempi di moderni sistemi tecnologici. In contesti industriali, molti sistemi tecnologici sono in realtà sistemi di sistemi costruiti integrando applicazioni indipendenti e modulari che, operando nel loro insieme, soddisfano bisogni specifici [11]. Esempi di sistemi tecnologici su larga scala includono mezzi navali, aerei ed impianti di trivellazione per l'estrazione di risorse naturali.

In genere, i sistemi tecnologici hanno tempi di vita lunghi, presentando potenzialmente un impatto significativo sul loro ambiente e sui loro utilizzatori. Alcuni di questi, come le centrali nucleari o i mezzi di trasporto, sono sistemi i cui malfunzionamenti potrebbero comportare ingenti perdite economiche ed umane. In letteratura ci si riferisce a tali sistemi con il nome di *critical systems*. Questi possono essere classificati in tre categorie, in base al loro scopo, al loro dominio applicativo e alle conseguenze di una loro avaria (*failure*):

- **Safety-critical:** sistemi la cui failure può comportare perdite di vita umana o ingenti danni ambientali (ad esempio, sistemi di controllo di apparecchiature medicali)
- **Mission-critical:** sistemi la cui failure può comportare il conseguente fallimento di un'attività mirata ad un obiettivo (ad esempio, sistemi di navigazione di satelliti)
- **Business-critical:** sistemi la cui failure può comportare perdite economiche (ad esempio, sistemi di gestione dei conti correnti bancari)

1.1.1 Dependability

I sistemi critici devono quindi per natura soddisfare un alto livello di sicurezza funzionale (*dependability*).

Informalmente, un sistema soddisfa il requisito di dependability quando si ripone un elevato grado di fiducia sulle funzionalità da esso espletate e quando si ritiene che i risultati da esso prodotti siano in linea con quanto previsto in fase di progettazione, senza il riscontro di effetti indesiderati e comportamenti non previsti. Ciò comporta, innanzitutto, che il termine dependability non assume un significato di valore fino a quando non sono resi espliciti tutti i principali rischi e il contesto in cui il sistema opera, tramite la definizione delle priorità dei requisiti del sistema ed una articolazione delle assunzioni di progetto. Gli effetti di un sistema critico si ripercuotono sul sistema stesso, sugli utenti e sull'ambiente in cui esso opera; di conseguenza la dependability dovrebbe essere intesa considerando tutti questi elementi (il contesto) e non può essere ridotta a proprietà locali, come la probabilità di crash o la conformità ad un dato protocollo o standard. In questi termini, gli utenti che interagiscono con il software dovrebbero essere visti non come entità esterne al confine (*boundary*) dello stesso, ma come parte integrante del sistema. Come conseguenza, diventa cruciale l'adozione di un approccio basato sull'ingegneria dei sistemi, in cui il software, componente principale dei sistemi critici, è visto come un artefatto ingegnerizzato in un sistema più complesso,

costituito da molti componenti, e in cui la ricerca della dependability rappresenta un fattore di equilibrio tra costi e benefici.

Sfortunatamente non esiste una definizione univoca di dependability né, a maggior ragione, un processo che ne valuti l'effettivo conseguimento in un sistema critico. Una tra le più utilizzate è la seguente [12]:

Dependability: “capacità di un sistema di fornire un servizio su cui possa essere riposta giustificata fiducia”.

Tale definizione comporta un'esposizione del concetto di dependability completa e lineare che consiste nell'analisi di tre fattori (Figura 1):

- **Attributi** (*attributes*): proprietà del concetto di dependability
- **Minacce** (*threats*): fattori che possono abbassare il grado di dependability di un sistema
- **Mezzi** (*means*): strumenti di mitigazione delle minacce della dependability

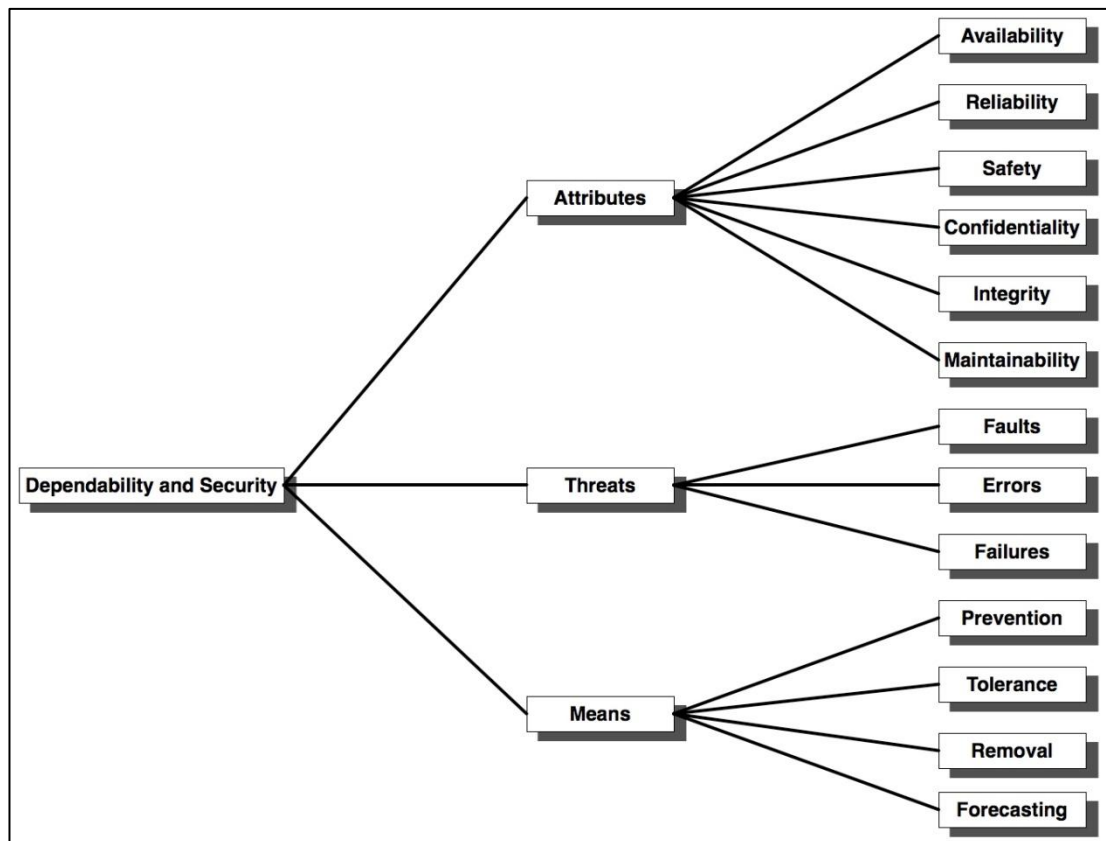


Figura 1: Tassonomia della dependability

A fronte di ciò, si definiscono i concetti di:

- **Utente di un sistema** (*system user*): altro sistema (umano o meno) che interagisce con il sistema attraverso un'interfaccia
- **Servizio** (*service*): funzionamento di un sistema, così come percepito da un utente
- **Funzione di un sistema** (*system functionality*): motivo per cui esso è stato progettato, descritta dalle specifiche di sistema.

1.1.1.1 Attributi

Come accennato, la dependability è un concetto ampio e complesso. Una sua completa definizione include i seguenti attributi di qualità:

- **Affidabilità** (*reliability*): continuità di un funzionamento corretto. Si esprime, fissando un intervallo di tempo e delle condizioni operative, come la probabilità che il sistema porti a termine il compito per cui è stato progettato.
- **Disponibilità** (*availability*): capacità di essere disponibile per l'erogazione di un servizio corretto. Si esprime come rapporto fra il tempo nel quale il sistema presenta un funzionamento corretto ed il tempo di esercizio totale.
- **Sicurezza** (*safety*): assenza di conseguenze catastrofiche per gli utenti e l'ambiente operativo. Si esprime come il tempo in cui un sistema si può considerare sicuro, evitando da tali conseguenze
- **Riservatezza** (*confidentiality*): prevenzione dalla diffusione non autorizzata d'informazioni.
- **Integrità** (*integrity*): prevenzione da alterazioni improprie dello stato del sistema
- **Manutenibilità** (*maintainability*): capacità di un sistema di essere sottoposto facilmente a modifiche e/o riparazioni

1.1.1.2 Minacce

Si dice che un sistema fornisce un *servizio corretto* quando quest'ultimo implementa la funzione del sistema. I problemi legati all'erogazione di un *servizio improprio* (non corretto) possono essere schematizzati attraverso la definizione di tre concetti principali:

- **Avaria o guasto** (*failure*): transizione dall'erogazione di un servizio corretto a un servizio non in linea con la funzione del sistema. Conseguenza dell'avaria è l'incidente (*accident*). La transizione inversa, cioè da un servizio improprio al servizio corretto è detta ripristino del sistema (*system restoration*)
- **Errore** (*error*): parte dello stato del sistema che può causare un successivo fallimento: un'avaria occorre quando si verifica un errore che altera un

servizio. Un errore è rilevato (*detected*) se la sua presenza è segnalata da un messaggio o un segnale di errore prodotto dal sistema. In caso contrario l'errore è detto latente (*latent*).

- **Difetto** (*fault*): causa, ipotizzata o meno, di un errore. Un difetto è attivo (*active*) quando produce un errore, in caso contrario è latente (*latent*). Un difetto in origine causa un errore all'interno di uno o più componenti del sistema, ma la failure del sistema non occorrerà fin quando l'errore non raggiungerà l'interfaccia di sistema, cioè non produrrà effetti visibili.

Le relazioni tra avarie, errori e difetti (*fault pathology*) sono riportate nella Figura 2.

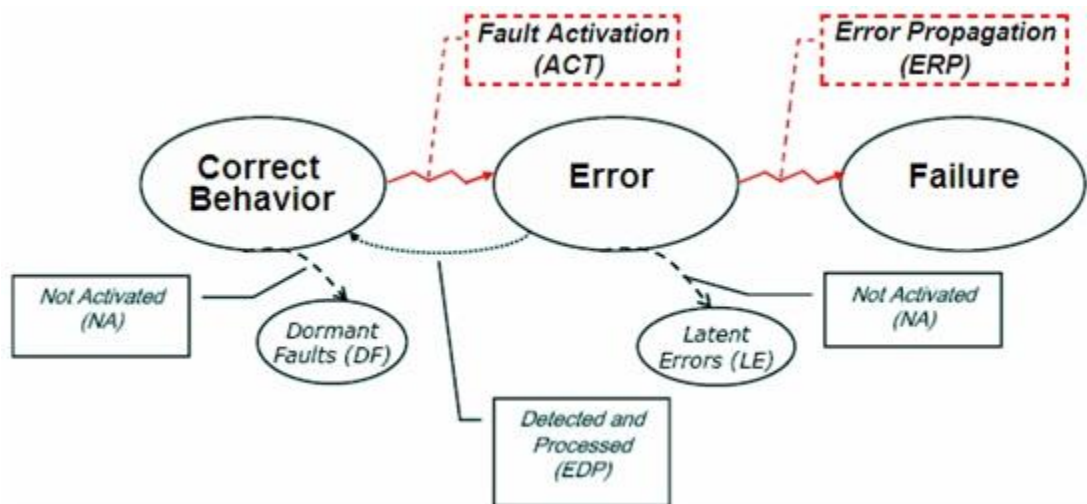


Figura 2: Relazioni tra avarie, errori e difetti

La rilevazione e la correzione di un errore (nato da un difetto attivo) portano allo stato di corretto funzionamento del sistema; in caso contrario, la propagazione fino all'interfaccia di sistema di un errore non latente porta a un'inevitabile failure.

1.1.1.3 Mezzi

Sono previsti diversi strumenti in grado di aumentare il grado di dependability del sistema, mitigando in vari modi le minacce appena descritte. La scelta del particolare approccio dipende dalla tipologia di sistema e dallo specifico attributo che si vuole migliorare.

- **Prevenzione dei difetti** (*fault prevention*): insieme di tecniche con cui prevenire l'occorrenza di difetti nel sistema. Questo risultato si ottiene attraverso tecniche di controllo della qualità durante la fase di design e realizzazione dell'hardware e del software.
- **Tolleranza ai guasti** (*fault tolerance*): insieme di tecniche con cui fornire un servizio corretto in presenza di difetti attivi. Esso è generalmente implementato da tecniche di rilevamento degli errori e successivo ripristino del sistema. Il rilevamento dell'errore origina un segnale o un messaggio di errore all'interno del sistema.
- **Rimozione dei difetti** (*fault removal*): insieme di tecniche con cui ridurre il numero o la gravità dei difetti durante la fase di sviluppo oppure durante la vita operativa di un sistema.
- **Previsione dei difetti** (*fault forecasting*): insieme di tecniche con cui stimare il numero di difetti, la loro evoluzione futura e le relative conseguenze. Tale previsione è condotta eseguendo una valutazione del comportamento del sistema in funzione delle occorrenze dei difetti, ossia dei difetti attivi

1.1.2 Rischio

La forte necessità di un sistema *dependable*, in congiunzione con le consistenti difficoltà di realizzazione, rende i sistemi tecnologici economicamente costosi da sviluppare e mantenere. Nonostante esistano concrete opportunità di benefici in relazione all'adozione di sistemi tecnologici, esistono dei rischi correlati al loro impatto e alle loro prestazioni.

Lo stesso concetto di dependability, infatti, prevede la nozione di sicurezza (*safety*). Anche la sicurezza di un sistema non possiede una definizione universale, tuttavia una tra le più accettate [13] riporta:

Safety: "libertà da un rischio inaccettabile di ferimento delle persone o di danni alla salute, direttamente o indirettamente, a seguito di un evento indesiderato"

Tale definizione però non è completa, ma dipende da una serie di concetti in relazione tra loro (Figura 3).

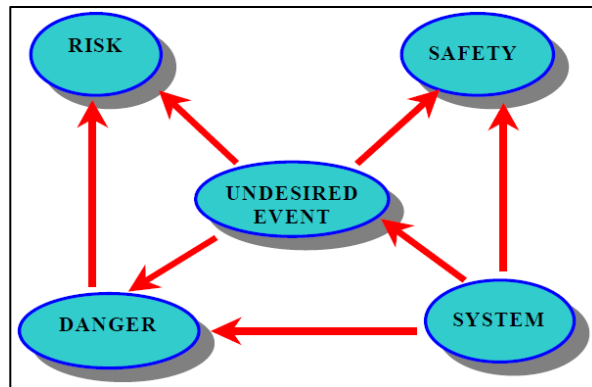


Figura 3: Relazione tra i concetti legati alla sicurezza

- **Evento indesiderato o incidente** (*undesired event* o *accident*): fenomeno imprevisto che può provocare il decesso di individui, lesioni, malattie professionali, oppure perdita o danni ad attrezzature, ai beni, all'ecosistema e all'impianto stesso.
- **Rischio** (*risk*): evento o una azione
 - associato ad una perdita
 - coinvolgente delle scelte
 - coinvolgente incertezze o pericoli.
- **Pericolo** (*danger*): condizione esistente o potenziale che può causare o contribuire ad un incidente.

Il concetto di incertezza relativa ad un rischio è relazionato con la conoscenza parziale di informazioni: se essa fosse completa, infatti, non vi sarebbe alcuna incertezza ed il rischio potrebbe essere determinato con precisione. Naturalmente, nello sviluppo di sistemi complessi, questa eventualità è utopica.

Più formalmente, è possibile definire il rischio come il prodotto della probabilità P che si verifichi un incidente per la gravità G delle sue potenziali conseguenze.

Se si considerano P e G come le uniche componenti di un evento indesiderato è possibile rappresentare il rischio in un piano cartesiano, definito spazio del rischio (*risk space*) e riportato in Figura 4.

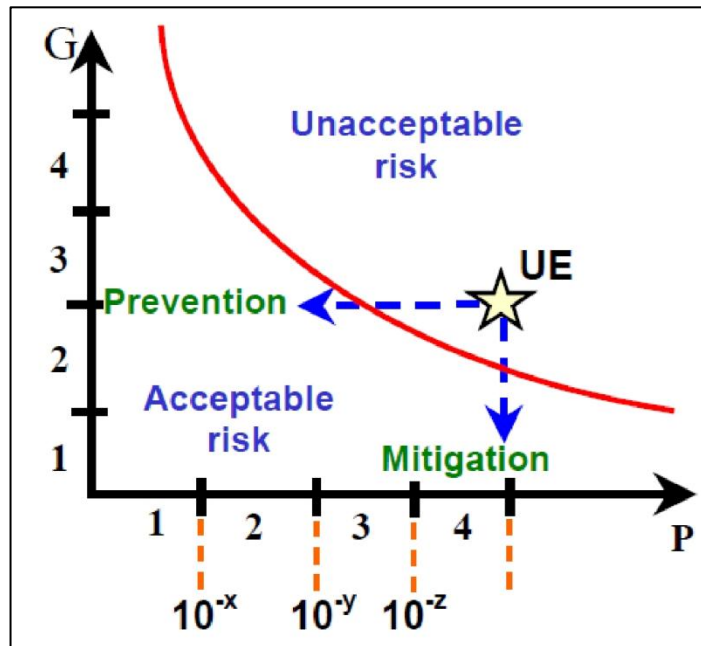


Figura 4: Spazio del rischio

È possibile osservare come esistano due modi per abbassare il rischio di incidente:

- **Attenuazione** (*mitigation*): rappresentata dalla freccia verticale, consiste nella diminuzione della gravità delle conseguenze dell'incidente.
- **Prevenzione** (*prevention*): rappresentata dalla freccia orizzontale, corrisponde alla diminuzione della probabilità che occorra che l'incidente.

L'abbassamento del rischio introduce altri due concetti complementari, riportati in Figura 5:

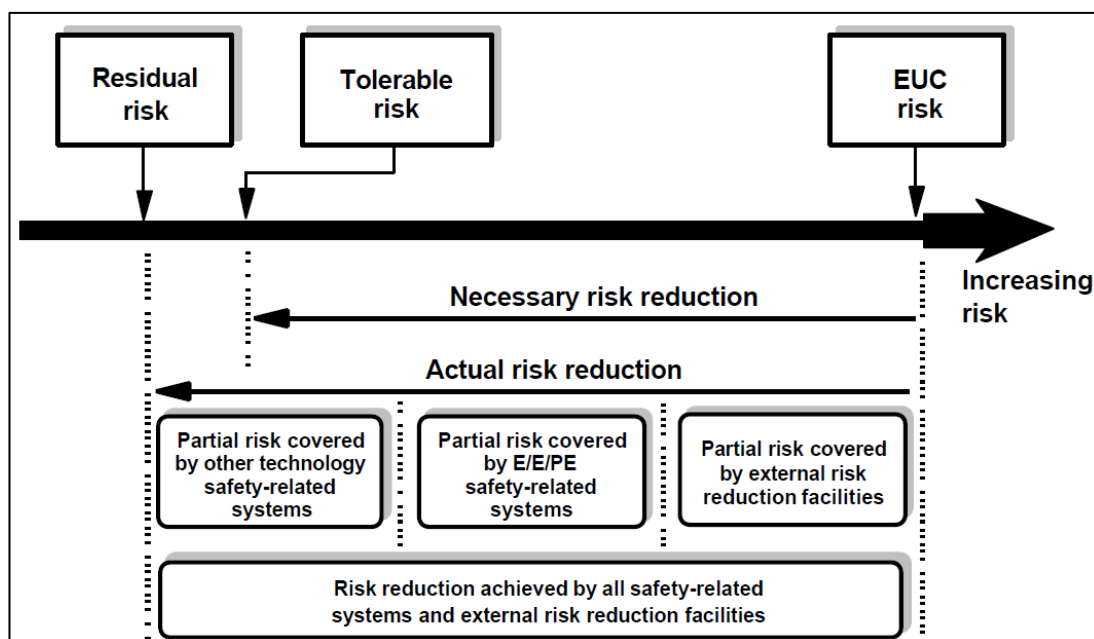


Figura 5: Concetti generali della riduzione del rischio

- **Riduzione del rischio al livello accettabile** (*necessary risk reduction*): porzione di rischio che viene eliminata attraverso la mitigazione o la prevenzione, che porta il rischio a livelli tollerabili (*tolerable risk*).
- **Rischio residuo** (*residual risk*): rischio marginale dell'evento indesiderato che si posiziona al di sotto della curva di rischio accettabile.

La fattorizzazione nel rischio in probabilità e gravità di incidenti è propria anche di numerosi standard di sicurezza [14], che definiscono, seppur con lievi differenze, una matrice del livello di rischio analoga a quella riportata in Figura 6.

Probability	Severity				
	Insignificant	Minor	Moderate	Significant	Catastrophic
Certain	MEDIUM	HIGH	EXTREME	EXTREME	EXTREME
Likely	MEDIUM	HIGH	HIGH	EXTREME	EXTREME
Possible	LOW	MEDIUM	HIGH	EXTREME	EXTREME
Unlikely	LOW	LOW	MEDIUM	HIGH	EXTREME
Remote	LOW	LOW	MEDIUM	HIGH	HIGH

Figura 6: Matrice del livello di rischio

1.1.2.1 Gestione del rischio

I rischi relativi ai sistemi tecnologici possono essere divisi in due categorie principali:

- **Rischi di sicurezza operativa** (*dependability risks*), derivanti da malfunzionamenti di servizio che sono più frequenti di quanto sia accettabile
- **Rischi di costi-benefici** (*cost-effectiveness risks*), posti da vincoli su tempi e budget realizzativi in termini di sottodimensionamento delle caratteristiche di un sistema a fronte dei suoi obiettivi reali

È ovvio come sia cruciale valutare e gestire ambedue i tipi di rischi durante l'intero ciclo di vita del sistema, dal suo concepimento fino alla manutenzione ed allo smantellamento.

A tal proposito, si definiscono i concetti di:

- **Gestione del rischio** (*risk management*), processo mediante il quale si misura o si stima il rischio e si sviluppano le strategie atte a governarlo.
- **Standard di sicurezza** (*security standard*), metodologia che definisce delle linee guida per la gestione del rischio.

La Figura 7 schematizza le attività fondamentali della gestione del rischio ed evidenzia l'impatto che queste hanno sul normale processo di sviluppo di un sistema.

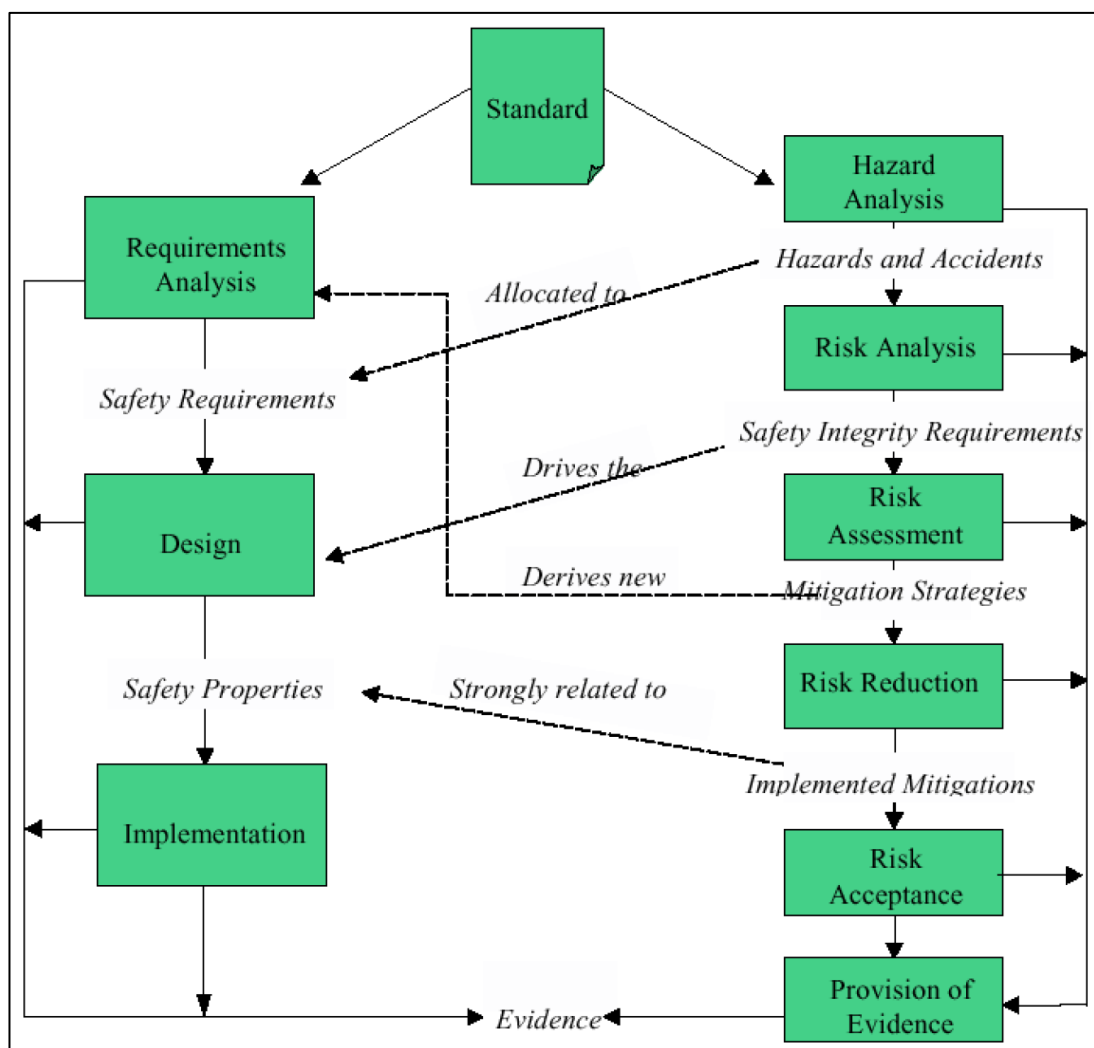


Figura 7: Interazioni fra il processo di sviluppo ed il processo di gestione dei rischi

Il processo di gestione del rischio è composto dalle seguenti attività:

- **Analisi dei pericoli** (*hazard analysis*), volta a produrre una lista di possibili pericoli ed incidenti
- **Analisi del rischio** (*risk analysis*), volta a stimare il livello di integrità del sistema che guida la progettazione del prodotto
- **Valutazione del rischio** (*risk assessment*), volta a definire le corrette strategie per mitigare e prevenire i rischi da cui sono derivati nuovi requisiti
- **Riduzione del rischio** (*risk reduction*), volta a definire le modalità operative per la prevenzione dei rischi

- **Accettazione del rischio** (*risk acceptance*), volta a stabilire un livello di rischio accettabile
- **Raccolta delle evidenze** (*provision of evidence*), volta a collezionare l'insieme di prove alla base dell'effettiva implementazione dello standard di sicurezza del sistema

Una pratica comune per molti tipi di sistemi è, infatti, disporre, al termine dell'attività di gestione del rischio, di un *documented case*, cioè di un insieme di argomentazioni sull'effettiva mitigazione del rischio. Tali argomentazioni devono essere supportate da evidenze oggettive ed incontestabili: è, infatti, sempre più comune negli ultimi tempi [15] sviluppare dei *safety case* per dimostrare che gli obiettivi di sicurezza del sistema sono stati raggiunti entro ragionevoli limiti.

1.2 Qualificazione di tecnologie

La valutazione (*assessment*) dell'effettivo soddisfacimento degli obiettivi di un sistema in termini di mitigazione dei rischi è un compito difficile quanto più il sistema è complesso. Una delle pratiche più comuni in merito è la certificazione (*certification*) condotta da un ente regolatore universalmente riconosciuto, che consiste nella verifica che un sistema è conforme ad uno standard di sicurezza.

Tuttavia, in settori in rapido sviluppo, come quello energetico, gli enti di certificazione devono esaminare sistemi facenti uso di tecnologie innovative per cui non esistono ancora degli standard. Si consideri, a tal proposito, il seguente caso di studio [16] riguardante il dominio applicativo delle piattaforme petrolifere in mare aperto.

Per diverso tempo le catene di acciaio sono state utilizzate come mezzo principale di ormeggio per strutture sottomarine e galleggianti. Di recente, si è manifestato un interesse sempre

crescente verso tecnologie in funi di fibra, sia in alternativa all'acciaio che per permettere di svolgere operazioni prima impossibili, come installazioni sott'acqua. Gli standard esistenti per ormeggi ed installazioni si focalizzano però esclusivamente sulle catene d'acciaio. Dal momento che acciaio e fibre hanno proprietà meccaniche e fisiche molto diverse, questi non sono idonei in un simile scenario: gli standard esistenti non sono né applicabili completamente né tantomeno coprono l'intero insieme delle caratteristiche delle funi in fibra.

Per verificare che una nuova tecnologia funzioni secondo le modalità previste in fase di progettazione si procede quindi con un tipo di valutazione diverso dalla certificazione, che prende il nome di qualificazione di una tecnologia (*technology qualification*).

Lo scopo della qualificazione di una tecnologia è fornire una prova, entro accettabili livelli di confidenza, che una nuova tecnologia sarà operativa secondo obiettivi ed assunzioni previste: quest'attività è tanto più importante ed efficace quanto più la tecnologia è innovativa e non esistono pertanto standard di sicurezza in merito. La qualificazione verte principalmente sull'analisi di componenti, attrezzature ed assemblaggi, che possono essere definiti come nuove tecnologie. Essa offre benefici a tutti gli stakeholder di una tecnologia; nello specifico:

- Garantisce ai produttori, che lanciano la nuova tecnologia sul mercato, una prova di qualità e validità del loro prodotto.
- Garantisce agli integratori, che integrano la tecnologia in sistemi già esistenti, un mezzo per valutare l'effetto della nuova tecnologia sulla dependability di un sistema.
- Garantisce agli utenti finali, che devono ottimizzare i ritorni economici dei loro investimenti, un'informazione di alta qualità che gli stessi utenti possono impiegare per confrontare e scegliere una tecnologia tra diverse alternative.

1.2.1 Il processo di qualificazione secondo DNV

Det Norske Veritas (DNV) [3] è una fondazione internazionale indipendente di origine norvegese che opera in numerosi settori del secondario e del terziario e la cui principale competenza è valutare ed indirizzare la gestione del rischio.

DNV ha ideato un processo di qualificazione [17], riportato in Figura 8, che consiste di sei attività principali:

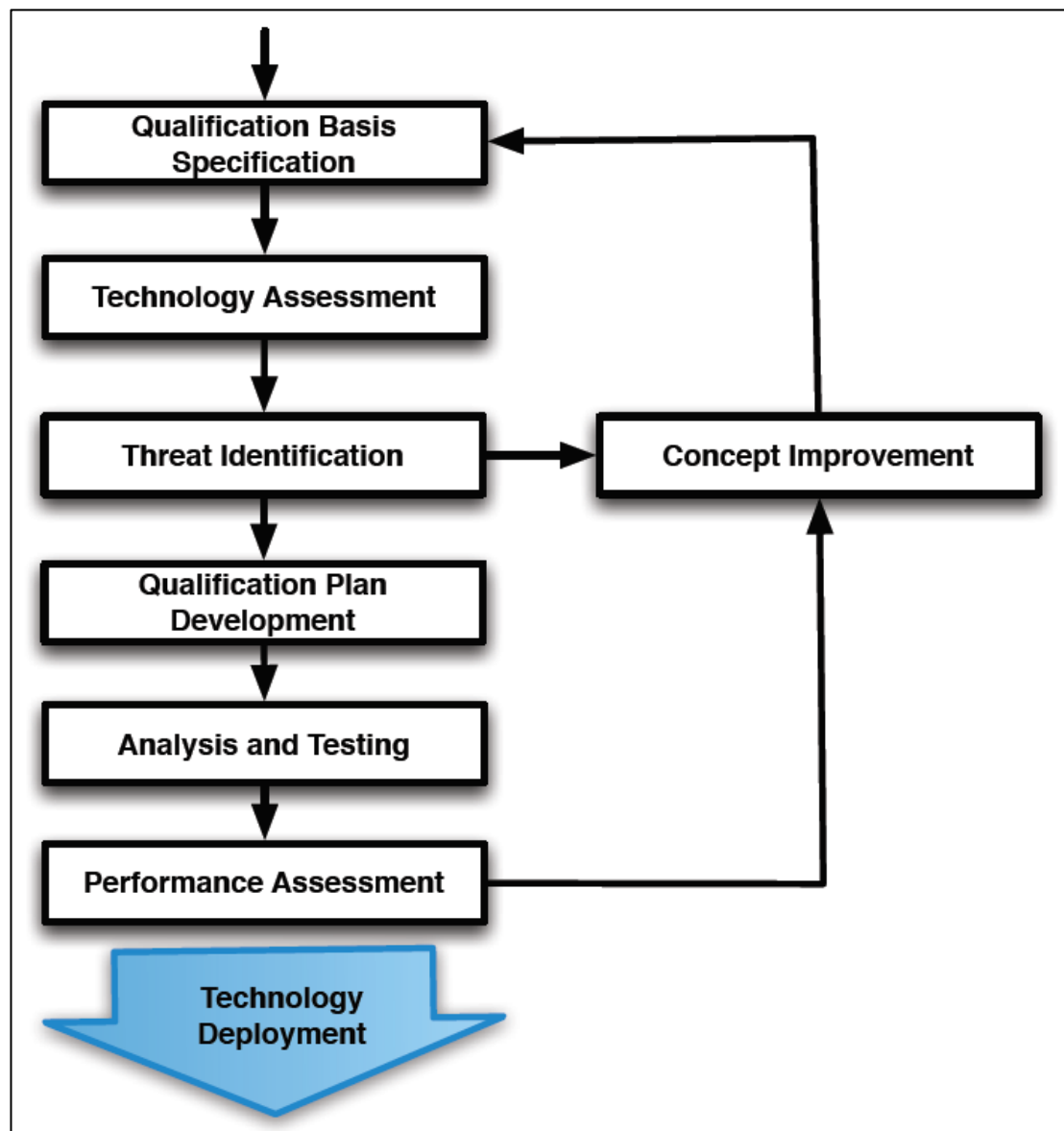


Figura 8: Il processo di qualificazione secondo DNV

- **Specifica delle basi della qualificazione** (*qualification basis specification*): sviluppo di una specifica per la descrizione della nuova tecnologia da qualificare in termini dei suoi requisiti di sicurezza, parametri principali, target di dependability e limiti operativi per trasporto, installazione, funzionamento e smantellamento
- **Valutazione della tecnologia** (*technology assessment*): divisione del sistema in parti componenti e valutazione delle singole novità introdotte nelle sue parti. Il *focus* dell'attività è sugli elementi componenti della nuova tecnologia e sull'identificazione dei suoi punti nevralgici
- **Identificazione delle minacce** (*threat identification*): creazione dei potenziali modelli di fallimento delle operazioni del sistema. Per le nuove tecnologie, non si dispone in genere di dati storici. È pertanto necessario determinare la dependability di un sistema attraverso la comprensione e l'analisi di un suo modello. La tecnologia deve pertanto essere decomposta ad un livello di dettaglio tale per cui ogni possibile modalità di fallimento può essere individuata e compresa. Quest'attività viene spesso condotta in seminari con ingegneri qualificati, che rappresentano esperti di dominio nelle discipline più rilevanti alla base della tecnologia. Ogni modalità di fallimento è classificata qualitativamente in base a conseguenze e probabilità dell'avvenimento, la cui combinazione determina il rischio complessivo (Sezione 1.1.2). Al termine dell'operazione, le modalità sono ordinate e priorizzate in base al loro coefficiente di rischio
- **Sviluppo del piano di qualificazione** (*qualification plan development*): definizione di un piano per la raccolta delle evidenze. I dati da raccogliere possono includere eventuali dati storici, analisi risultati di analisi tecniche, test in laboratorio, cambiamenti procedurali volti ad evitare potenziali problemi e test per ridurre l'incertezza in modelli analitici
- **Analisi e test** (*analysis and testing*): svolgimento effettivo delle attività di qualificazione pianificate. Queste attività in genere richiedono la maggior parte del budget assegnato al processo di qualificazione. Il compito più

difficile è ottenere informazioni quantitative, cioè condurre degli studi che possano documentare in termini matematici la probabilità di avvenimento di guasti.

- **Valutazione delle prestazioni** (*performance assessment*): quantificazione della dependability complessiva del sistema. L'obiettivo di quest'attività è confermare che i requisiti di sicurezza definiti nella fase iniziale di specifica siano effettivamente soddisfatti. Sebbene si possa provare che una determinata tecnologia sia adatta ad un servizio anche senza l'uso di misure di dependability, una valutazione quantitativa rende la stessa più competitiva in un mercato dove la redditività di una tecnologia è giocoforza direttamente correlata con la sua sicurezza funzionale
- **Evoluzione del processo** (*concept improvement*): attuazione di cambiamenti ed interventi migliorativi nel processo per poter venire incontro ad eventuali modifiche apportate alla tecnologia dettate da necessità di dependability. Nonostante sia stato presentato in forma lineare, il processo DNV è iterativo: ciò comporta che prima dell'effettiva messa in funzione, il *concept* di una tecnologia può subire diversi cambiamenti e miglioramenti sulla base dei risultati osservati durante le diverse fasi dell'attività di qualificazione.

In genere, le prime fasi del processo sono attuate dal produttore della tecnologia, che delega ad un'organizzazione di terze parti (come DNV stessa), le attività di analisi e valutazione delle prestazioni. Quest'ultimo compito può risultare impegnativo, a causa della necessità di dimostrare dei collegamenti tra le basi della qualificazione, i rischi identificati ed i test condotti.

1.3 Motivazioni

La qualificazione di nuove tecnologie è un'attività complessa e delicata, che coinvolge numerose figure professionali e dai risultati di considerevole valore. Sulla

base di studi condotti sulla letteratura attinente [18], è stata stilata una lista delle principali difficoltà che si incontrano nell'attuazione del processo di qualificazione, descritte nelle Sezioni 1.3.1-1.3.6. Queste sono state di spunto per la definizione di una nuova metodologia per la qualificazione, descritta nel Capitolo 2, e conseguentemente per lo sviluppo di uno strumento a supporto di tale metodologia, descritto nel Capitolo 3. Sebbene sia difficile incorrere simultaneamente in tutti i problemi menzionati, una lista aggregata è utile per l'inquadramento delle motivazioni alla base dello studio oggetto della presente tesi di laurea. Nello specifico, sono già esistenti delle soluzioni parziali ad alcune problematiche, e futuri sviluppi in questa direzione avranno, tra gli altri, l'obiettivo di studiarle, generalizzarle e condensarle all'interno di un unico *framework* operativo.

1.3.1 Scarsa organizzazione delle informazioni necessarie per la valutazione

- **Problema:** gli obiettivi generali riguardo il grado di soddisfazione del sistema non sono sistematicamente decomposte in argomentazioni a grana più fine e più gestibili, per le quali è possibile raccogliere delle evidenze concrete
- **Cause:** non esistono linee guida chiare su come modellare le relazioni tra conclusioni, argomentazioni ed evidenze a supporto
- **Effetti:** saranno inevitabili delle ripercussioni negative sul grado di soddisfacimento degli obiettivi di un sistema. In particolare, la raccolta delle evidenze e l'elicitazione delle opinioni degli esperti sono efficaci solo per argomenti a grana fine
- **Soluzione proposta:** definizione di una metodologia fortemente allineata con la letteratura in tema di analisi *goal-oriented* nel campo dell'ingegneria dei requisiti, rendendo immediatamente applicabili formalismi già esistenti.

1.3.2 Tardiva raccolta delle evidenze

- **Problema:** durante il processo di sviluppo del sistema, le evidenze richieste per la valutazione non sono raccolte abbastanza tempestivamente, o sono raccolte successivamente in un'attività sperata
- **Cause:** non esiste in genere un'accordo a priori riguardo l'esatto contenuto e struttura delle evidenze che devono essere raccolte, e su come queste evidenze dovrebbero essere collegate agli obiettivi
- **Effetti:** una raccolta tardiva delle evidenze è spesso costosa in termini di risorse temporali e soggetta ad errori
- **Soluzione proposta:** definizione ed analisi degli obiettivi degli stakeholder sin dai primi stadi del ciclo di vita dello sviluppo, in modo che gli sviluppatori stessi possano decidere quanto prima su quali sono le evidenze necessarie per la valutazione.

1.3.3 Modalità e meccanismi di fallimento non specificati

- **Problema:** l'insieme identificato delle modalità e dei meccanismi di fallimento è spesso incompleto
- **Cause:** è difficile identificare esattamente gli eventi alla base dei fallimenti prima di un'adeguata decomposizione (strutturale o funzionale) del sistema
- **Effetti:** non riuscire ad identificare correttamente le modalità ed i meccanismi di fallimento significa non riuscire a stabilire correttamente il rischio nel processo di valutazione, rendendo impossibile il soddisfacimento degli obiettivi del sistema
- **Soluzione proposta:** definizione di un formalismo che aiuti ad identificare gli ostacoli al raggiungimento degli obiettivi del sistema, integrato in tecniche già esistenti come *FMEA* e *FTA* [19]

1.3.4 Inadeguato supporto decisionale per il miglioramento del sistema

- **Problema:** quando la valutazione stabilisce che un obiettivo non è soddisfatto, è difficile determinare quali aspetti del sistema necessitino di miglioramenti, ed in che misura dovrebbero essere apportati dei cambiamenti
- **Cause:** i fattori che impattano significativamente sul soddisfacimento di un obiettivo non possono essere facilmente individuati
- **Effetti:** potrebbero verificarsi sforamenti di tempi e budget previsti, in quanto l'effort non è focalizzato sul miglioramento degli aspetti rilevanti del sistema
- **Soluzione proposta:** definizione di meccanismi per misurare la sensibilità di un obiettivo a diversi fattori considerati durante la valutazione, e per determinare la misura in cui attuare miglorie e cambiamenti per soddisfare uno specifico obiettivo.

1.3.5 Inadeguato supporto per l'identificazione dell'impatto dei cambiamenti

- **Problema:** obiettivi dettagliati e componenti fondamentali di un sistema sono spesso soggetti a cambiamento a causa dell'evoluzione di necessità di business. Al momento non ci sono soluzioni effettive per misurare l'impatto dei cambiamenti sul soddisfacimento degli obiettivi di un sistema
- **Cause:** le relazioni tra obiettivi e le argomentazioni ed evidenze a loro supporto non sono definite abbastanza chiaramente da poter consentire dei ragionamenti riguardo la propagazione dei cambiamenti attraverso il sistema
- **Effetti:** una stima inaccurata dell'impatto dei cambiamenti può portare a due tipi di problemi. In prima istanza, un cambiamento che apporti miglorie senza avere impatti significativamente negativi sugli obiettivi del sistema

può essere rifiutato. Al contrario, un cambiamento con un significativo impatto negativo può essere accettato, minacciando budget, tempi e dependability del sistema

- **Soluzione proposta:** inclusione nella metodologia di un efficace meccanismo per la stima dell'impatto dei cambiamenti effettuati sugli obiettivi di livello più basso sugli obiettivi di livello più alto.

1.3.6 Trattamento implicito dell'ambiente operativo

- **Problema:** le proprietà dell'ambiente in cui opera la tecnologia non sono definite esplicitamente. Ciò complica notevolmente lo studio sulla dependability, in quanto esso richiede un'esplicita articolazione delle assunzioni ambientali [18]
- **Cause:** l'impatto del contesto applicativo di una tecnologia viene spesso sottostimato: ciò avviene soprattutto quando il sistema che ne farà utilizzo è estensivamente *software-based*
- **Effetti:** la sicurezza funzionale del sistema viene impattata negativamente, perché le condizioni sull'ambiente operativo assunte implicitamente in fase di sviluppo possono non verificarsi al momento dell'attuazione pratica, e pertanto gli obiettivi potrebbero dimostrarsi non più soddisfacenti. Inoltre, i risultati di un'attività di qualificazione basata su assunzioni non esplicite possono essere difficilmente riutilizzabili in altri contesti
- **Soluzione proposta:** inclusione nella metodologia di meccanismi atti ad identificare le assunzioni esplicitamente, per poterne considerare gli effetti durante la stima del soddisfacimento degli obiettivi.

2 Modus: un approccio innovativo per la qualificazione di nuove tecnologie

FAV_QUOTE_PLACEHOLDER

2.1 Visione d'insieme

Per far fronte alle difficoltà che attualmente si presentano nell'attività di qualificazione di nuove tecnologie (Sezione 1.3), il *Simula Research Laboratory* di Oslo [2] ha sviluppato una metodologia innovativa per la qualificazione delle nuove tecnologie che prende il nome di *Modus* [1].

I problemi riscontrati negli attuali processi di qualificazione evidenziano che l'utilizzo di modelli che rappresentassero graficamente le entità coinvolte avrebbe migliorato sensibilmente efficienza e chiarezza dell'attività di valutazione, fornendo un framework all'interno del quale presentare i risultati finali in modo trasparente. L'introduzione di una metodologia *model-based* in un processo di qualificazione comporterebbe pertanto notevoli benefici, potendo essa essere utilizzata sin dalle prime fasi per descrivere rischi, requisiti ed attività: al termine del processo si potrebbe infatti chiaramente risalire a tutte le operazioni che hanno coinvolto un determinato oggetto.

Una caratteristica fondamentale di Modus è la sua natura *goal-oriented*: ciò comporta la definizione di un processo di qualificazione guidato dagli obiettivi che

una nuova tecnologia deve soddisfare. Un obiettivo (*goal*) è definibile come un’asserzione descrittiva di un proposito di una tecnologia [9]. Gli obiettivi possono essere espressi a diversi livelli di astrazione: obiettivi di alto livello sono in genere direttive strategiche come “Assicurare la sicurezza”, “Espandere i mercati” o “Aumentare i profitti”. Obiettivi di livello più basso sono invece più specifici ed in genere rappresentano requisiti tecnici legati alla progettazione come “L’allarme antincendio deve scattare entro 2 secondi dall’incendio stesso”.

La Figura 9 riporta una visione d’insieme di Modus, specificando le attività in esso definite ed effettuando un parallelismo con il processo definito da DNV (Sezione 1.2.1).

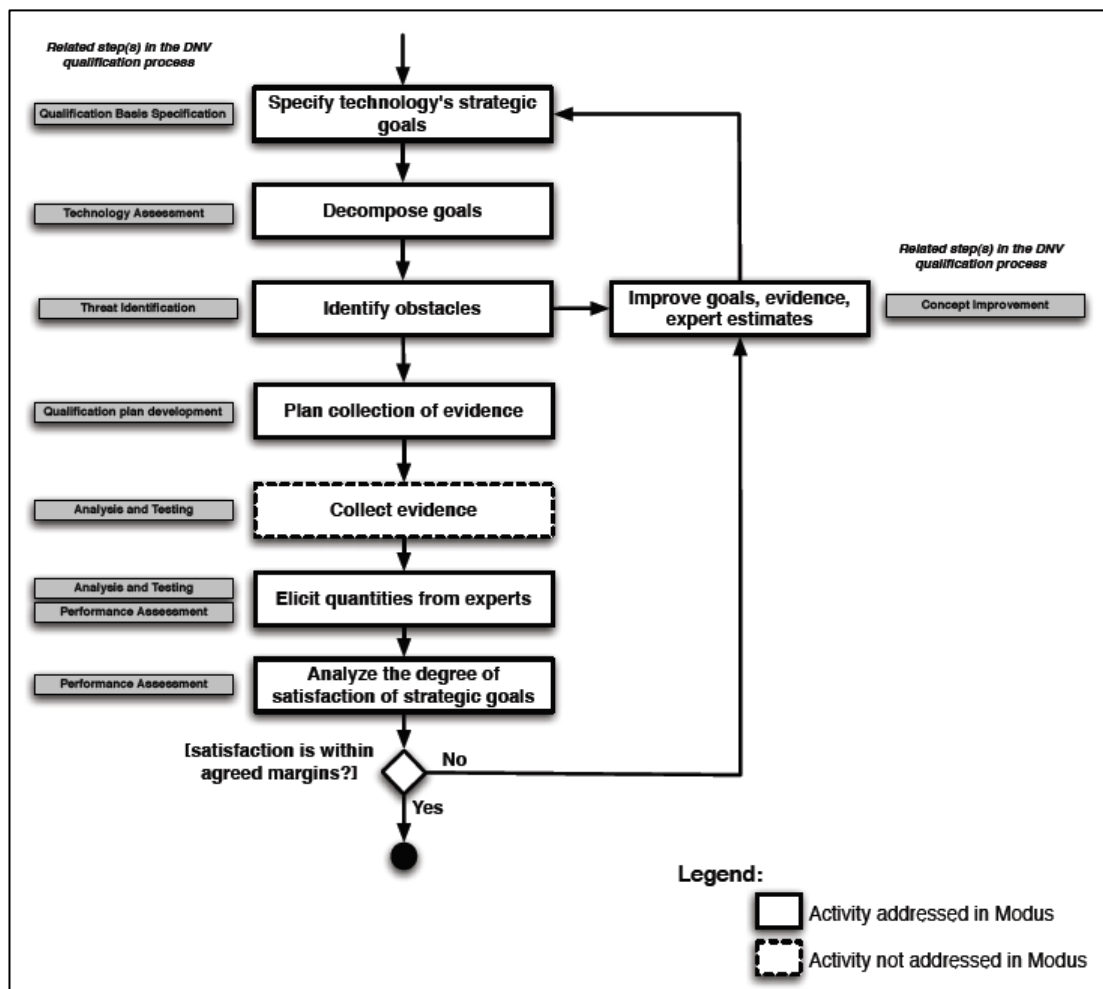


Figura 9: Visione d’insieme della metodologia Modus

La metodologia si articola in 8 fasi:

- Specifica degli obiettivi strategici della tecnologia (*specify technology's strategic goals*): il primo passo della metodologia è specificare gli obiettivi strategici che la nuova tecnologia deve perseguire. Questa fase corrisponde alla “specificazione delle basi della qualificazione” secondo il modello DNV.
- Decomposizione degli obiettivi (*decomposition of goals*): gli obiettivi strategici devono essere decomposti in obiettivi più specifici relativi ai vari componenti della tecnologia, quali possono essere attrezzature meccaniche, elettriche, software o risorse umane. Questa decomposizione deve aver luogo come parte della “valutazione della tecnologia” secondo il modello DNV.
- Identificazione degli ostacoli (*identification of obstacles*): la decomposizione degli obiettivi è seguita dall'identificazione di ostacoli che possono negare il soddisfacimento degli obiettivi stessi. L'identificazione di tali ostacoli può essere compiuta al meglio come parte dell'“identificazione delle minacce” secondo il modello DNV.
- Pianificazione della raccolta delle evidenze (*planning the evidence collection*): questa fase mira a sviluppare una strategia per la raccolta delle evidenze. Le evidenze supporteranno le conclusioni riguardo il soddisfacimento degli obiettivi, così come frequenza e gravità degli ostacoli. Quest'attività deve essere condotta come parte dello “sviluppo del piano di qualificazione” secondo il modello DNV.
- Raccolta delle evidenze (*collection of evidences*): questa fase è incentrata sulla raccolta delle evidenze pianificata al passo precedente. Modus non ha l'obiettivo di fornire delle linee guida per la raccolta e la strutturazione delle evidenze, dal momento che quest'attività dipende fortemente dal contesto applicativo, così come da standard e norme applicabili al sistema che si sta sviluppando.
- Elicitazione delle quantità dagli esperti (*elicitation of quantities from experts*): in questa fase, si chiede agli esperti di esprimere la loro opinione su

diverse quantità, come la probabilità che siano soddisfatti obiettivi (o si presentino ostacoli) di basso livello, così come il grado di completezza decompositiva di obiettivi ed ostacoli di livello più alto. Per essere significativa, l'opinione degli esperti deve essere elicitata utilizzando processi sistematici: occorre infatti minimizzare gli effetti di eventuali errori di elicitazione (*bias*) per assicurarsi che i dati siano quanto più possibile accurati. L'elicitazione delle opinioni degli esperti è condotta principalmente durante la fase di "analisi e testing" e nei primi stadi della "valutazione delle prestazioni" secondo il modello DNV.

- Analisi del soddisfacimento degli obiettivi strategici (*analyzing the satisfaction of strategic goals*): questa è una fase automatizzata, in cui i dati relativi al grado di soddisfacimento di obiettivi di livello più basso sono propagati agli obiettivi di livello più alto. La propagazione deve considerare che le opinioni degli esperti spesso contengono incertezze che devono essere propagate a loro volta. Questa fase va condotta come parte della "valutazione delle prestazioni" secondo il modello DNV.
- Miglioramento di obiettivi, evidenze e stime degli esperti (*improving of goals, evidence, and expert estimates*): per varie ragioni, potrebbero essere necessari miglioramenti e correzioni nell'attuazione del processo di qualificazione. Per esempio, si potrebbe aver bisogno di sviluppare delle contromisure per mitigare gli ostacoli, che introdurrebbero dei nuovi obiettivi, oppure dei cambiamenti in obiettivi esistenti. La stessa analisi del soddisfacimento degli obiettivi potrebbe essere la causa di cambiamenti: si potrebbe non essere soddisfatti del grado di soddisfacimento degli obiettivi, che seppur presente potrebbe essere troppo incerto. Sarebbe, quindi, richiesta dapprima l'identificazione degli obiettivi di basso livello che hanno un impatto significativo sul soddisfacimento degli obiettivi di alto livello e, in seconda istanza, l'attuazione di interventi di miglioramento, come l'utilizzo di componenti più efficaci o dependable, la raccolta di ulteriori evidenze e l'utilizzo di esperti più qualificati per l'elicitazione. Tutti questi possibili

miglioramenti avvengono nella fase di “evoluzione del processo” secondo il modello DNV.

Tutte le fasi di Modus, ad eccezione della raccolta delle evidenze, attività che non rientra nello scope della metodologia, sono esaminate nel dettaglio nelle Sezioni 2.2-2.8.

2.2 Specifica degli obiettivi strategici della tecnologia

Ad alti livelli di astrazione, gli obiettivi strategici di una tecnologia sono in genere noti agli stakeholder, ma nella pratica si rischia spesso di non esplicitarli durante l’esecuzione del processo di qualificazione. Modus propone una tempestiva esplicitazione di tali obiettivi, in modo da rendere possibile la loro decomposizione in sotto-obiettivi (*subgoal*) più dettagliati.

Una tecnica semplice ma efficace, che può essere utilizzata per l’identificazione degli obiettivi strategici, impiega tecniche di *keyword analysis* nei documenti preliminari e nelle specifiche di sviluppo della tecnologia oggetto di qualificazione. Un obiettivo, infatti, è per definizione un’asserzione in merito ad un proposito. Molti obiettivi, inoltre, prescrivono qualche miglioramento a tecnologie già esistenti. La Tabella 1, adattata da [9], riporta delle parole chiave di aiuto per la ricerca degli obiettivi nei documenti preliminari di una tecnologia.

Tipo	Keyword
Prescrittive	deve, non può mai, non può, non dovrebbe, non dovrebbe mai
Intenzionali	per poter, al fine di, obiettivo, scopo, ottenere, mantenere, evitare, assicurare, garantire, volere, aspettarsi
Migliorative	migliorare, aumentare, diminuire, ridurre, potenziare, supportare, abilitare,

fornire

Tabella 1: Parole chiave per la definizione degli obiettivi

Ad esempio, l'analisi di frasi, come "la tecnologia deve ridurre i costi operativi del 20%", "la tecnologia deve essere ragionevolmente sicura entro i suoi limiti operativi", porterebbe all'introduzione nel modello di obiettivi strategici.

Nonostante sia semplice ed utile, una ricerca non intelligente delle parole chiave all'interno di documenti ha le sue limitazioni. In particolare, potrebbero essere riscontrati numerosi falsi positivi, che dovrebbero essere filtrati dall'intervento umano. Inoltre, a meno che i documenti non presentino esclusivamente un linguaggio tecnico molto ristretto, potrebbe essere difficile definire un insieme esaustivo di tutte le parole chiave. Non tutti gli obiettivi strategici possono quindi essere identificati automaticamente: potrebbero, infatti, presentarsi anche dei falsi negativi. Identificare eventuali omissioni di obiettivi richiederebbe ancora una volta un intervento umano.

Nonostante le limitazioni, la ricerca di parole chiave si è dimostrata essere un'euristica molto performante per aiutare nell'individuazione di obiettivi ad alto livello.

2.3 Decomposizione degli obiettivi

La modalità standard di elaborazione degli obiettivi strategici è la divisione in sotto-obiettivi (*goal decomposition*). La decomposizione degli obiettivi continua finché non si giunge a sotto-obiettivi a grana così fine da consentire una raccolta diretta di evidenze a supporto del loro soddisfacimento (*leaf goal*). La decomposizione degli obiettivi è mirata a proporre una soluzione alle problematiche descritte nelle Sezioni 1.3.1 e 1.3.2.

Esistono diverse notazioni per descrivere la decomposizione degli obiettivi, tra cui il *Non-Functional Requirements Framework (NFR)* [20], *KAOS* [9], *i** [21], e *GSN* [15]. Tra questi, Modus utilizza la notazione KAOS in quanto essa fornisce una base rigorosa per il ragionamento quantitativo in merito al soddisfacimento degli obiettivi.

2.3.1 Decomposizione degli obiettivi secondo la notazione KAOS

Per illustrare la decomposizione degli obiettivi in KAOS, verrà utilizzato un semplice esempio, adattato da [9], dove viene decomposto un obiettivo strategico riguardante la sicurezza dei passeggeri di un treno. La Figura 10 riporta un diagramma di decomposizione dell'esempio secondo la notazione KAOS.

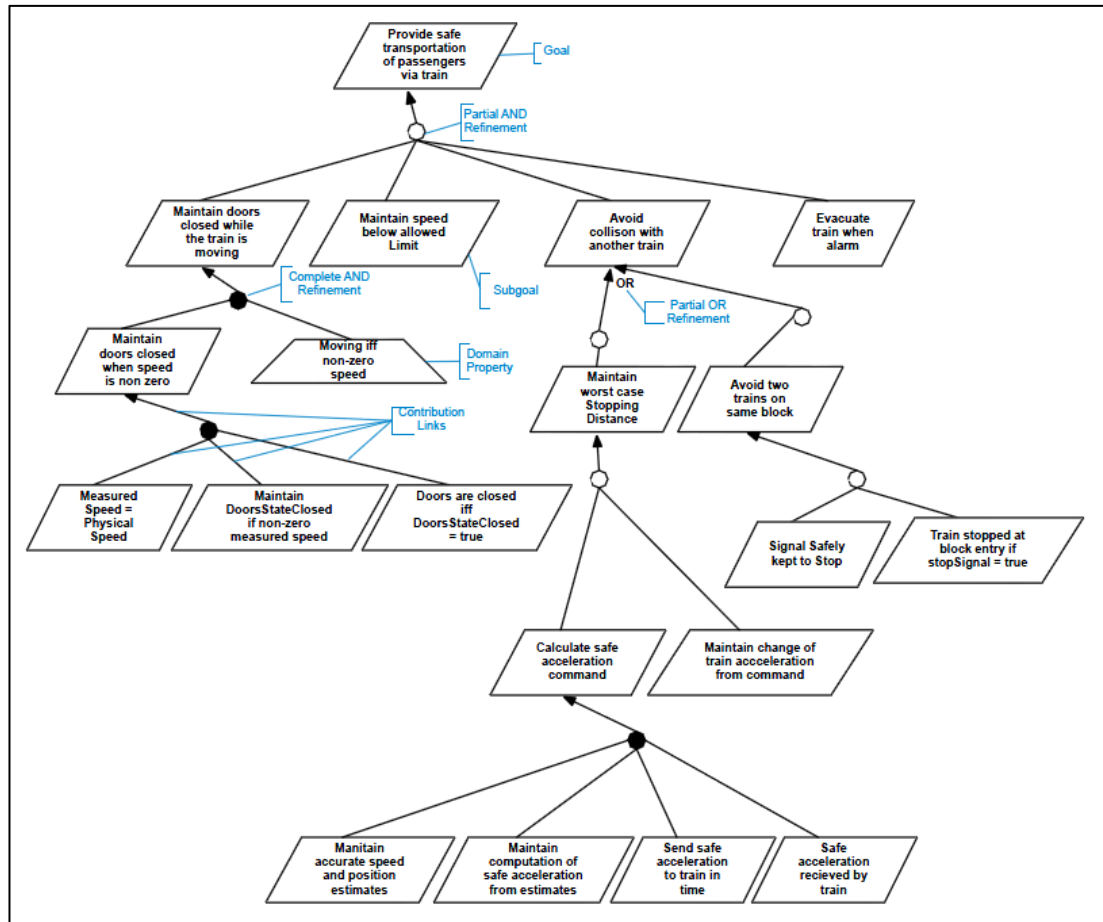


Figura 10: Decomposizione degli obiettivi in KAOS

Obiettivi e sotto-obiettivi sono rappresentati da parallelogrammi. Proprietà di dominio riguardanti gli obiettivi (*assumption*), cioè asserzioni descrittive riguardo l'ambiente del sistema ritenute vere indipendentemente dalle condizioni operative, sono rappresentati da trapezi. Esplicitare le assunzioni di un dominio applicativo mira a risolvere la problematica descritta nella Sezione 1.3.6.

La decomposizione degli obiettivi avviene utilizzando gli operatori *AND* ed *OR* per mostrare come diversi sotto-obiettivi contribuiscono insieme al soddisfacimento del loro obiettivo padre. Le decomposizioni possono essere complete o parziali: una decomposizione completa (*complete decomposition*) implica che un goal padre è stato completamente decomposto nei suoi sotto-obiettivi e non è possibile aggiungerne di altri. Una decomposizione parziale (*partial decomposition*) implica invece che la stessa non è esaustiva, ed ulteriori sotto-

obiettivi potrebbero essere aggiunti in futuro. Le decomposizioni complete sono indicate da cerchi pieni, mentre le decomposizioni parziali da cerchi vuoti.

Studiare il soddisfacimento degli obiettivi raramente porta a risposte di tipo binario ("sì"/"no"), motivo per cui è opportuno parlare di grado di soddisfacimento. Le sfumature in questo senso sono causate dalle incertezze che inevitabilmente si presentano nel processo di sviluppo, tra cui verifiche e validazioni parziali, livello di confidenza degli esperti e criteri, come la competenza dei team di lavoro. Un framework di valutazione quantitativa deve fornire soluzioni a questa problematica: Modus adotta infatti un approccio di tipo probabilistico per gestire il grado di incertezza nello studio delle composizioni (Sezioni 2.6 e 2.7).

2.3.2 Dimensioni della decomposizione degli obiettivi

L'esempio della Figura 10 riportava una decomposizione di un obiettivo di alto livello in termini del comportamento che deve o non deve seguire il sistema. Quella comportamentale (*behavioral decomposition*) non è l'unica dimensione di decomposizione degli obiettivi. Un altro approccio comune in merito è la decomposizione strutturale (*structural decomposition*) di un sistema. In alternativa, si potrebbe decomporre il processo di sviluppo, funzionamento o manutenzione di un sistema, piuttosto che le sue componenti (*process decomposition*). La Figura 11 riporta i tre tipi di decomposizione nell'esempio di un sistema missilistico adattato da [19].

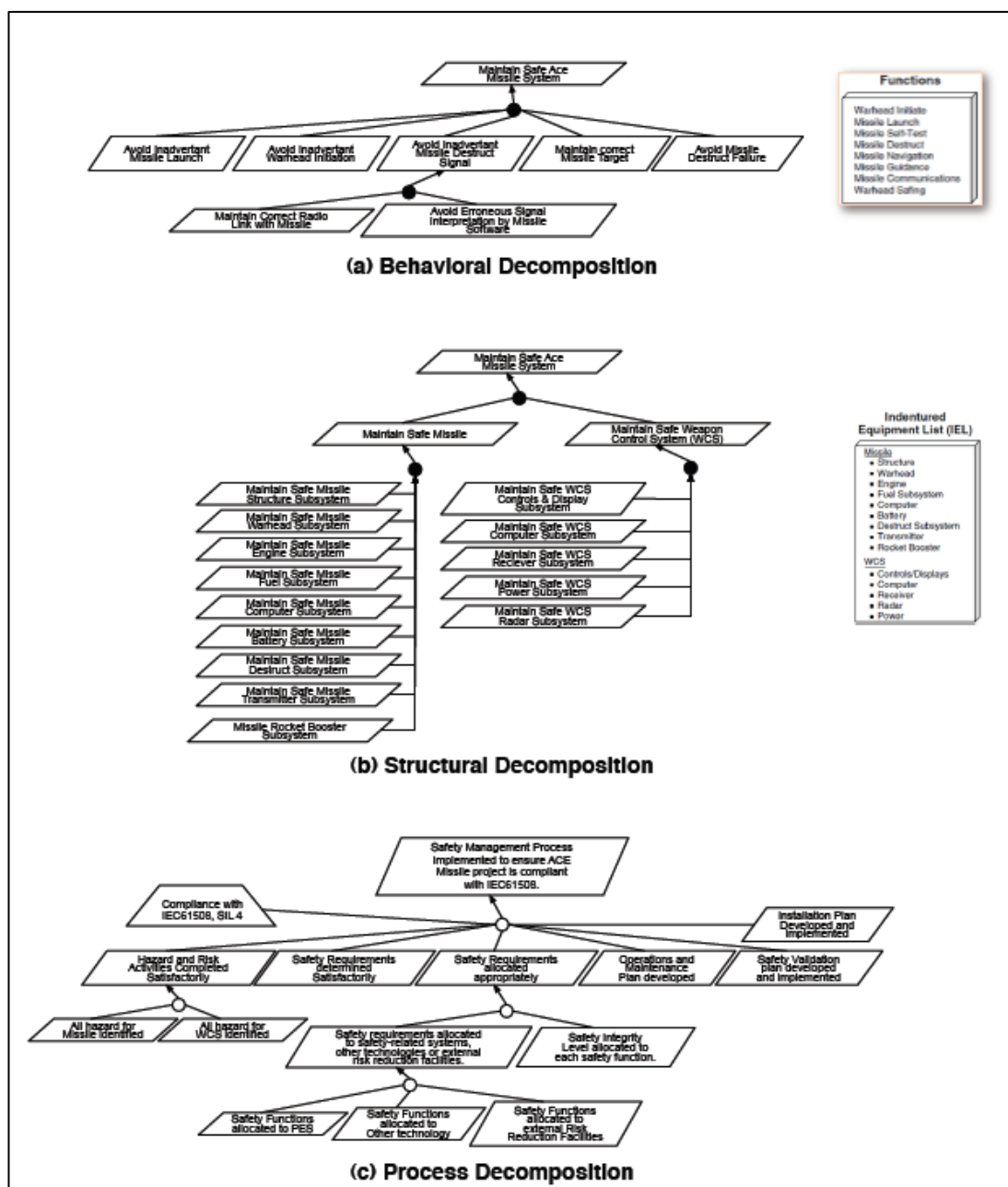


Figura 11: Dimensioni della decomposizione degli obiettivi

Sebbene non sia esplicitamente riportato nella figura, è possibile comporre diversi tipi di decomposizione in un singolo modello. Ad esempio, si potrebbe iniziare al primo livello con la decomposizione strutturale, per poi procedere su ogni componente con la decomposizione comportamentale. Allo stato attuale, comunque, Modus non fornisce linee guida per combinare diversi i tipi di decomposizione.

2.3.3 Euristiche di modellazione degli obiettivi

La ricerca di parole chiave, descritta nella Sezione 2.2 in merito alla specifica degli obiettivi strategici della tecnologia, non è l'unica euristica nota in letteratura [9] in merito allo studio degli obiettivi. Se ne descrivono di seguito le più rilevanti nell'ambito della qualificazione delle tecnologie:

- **Chiedersi “come” e “perché”:** dato un goal obiettivo *G*, si possono identificare sistematicamente i sotto-obiettivi e gli obiettivi padre di *G* chiedendosi due tipi di domande:
 - **Raffinamento di obiettivi attraverso “come”** (*HOW questions*): i sotto-obiettivi di *G* vengono identificati attraverso domande come: “come può essere soddisfatto *G*?”, “questo sotto-obiettivo è sufficiente oppure c'è bisogno di un altro obiettivo per soddisfare *G*?”
 - **Astrazione di obiettivi attraverso “perché”** (*WHY questions*): gli obiettivi padre di *G* vengono identificati attraverso domande come: “perché il sistema dovrebbe soddisfare *G*?”, “c'è un altro obiettivo padre a cui *G* contribuisce?”. Un pattern di domande utile in merito, consiste nel chiedere domande di tipo “perché” riguardo un obiettivo *G*, immediatamente seguite da domande di tipo “come” riguardo l'obiettivo padre trovato, in modo da individuare eventuali fratelli di *G* nella gerarchia di decomposizione.
- **Dividere le responsabilità tra componenti:** è possibile vincolare le domande di tipo “come” richiedendo che i sotto-obiettivi risultati abbiano meno componenti (risorse software, hardware o umane) richiesti nel loro soddisfacimento. Questa euristica è importante per assicurare che il processo di raffinamento progredisca verso un insieme di obiettivi concreti e a grana fine per cui sia possibile raccogliere delle evidenze concrete riguardo il soddisfacimento di questi.
- **Porsi domande di tipo “cosa-se”** (*WHAT-IF questions*): una volta identificato un obiettivo della forma “se si verifica una certa preconditione,

allora avviene un determinato effetto”, ci si potrebbe porre una domanda di tipo “cosa-se”, per controllare se sia necessario un obiettivo complementare che risponda alla domanda “cosa si verifica se non si verifica una determinata preconditione?”.

2.4 Identificazione degli ostacoli

In questa fase sono identificati gli ostacoli (*obstacle*) che prevengono (*obstruct*) il soddisfacimento degli obiettivi. Gli ostacoli vengono in genere rappresentati in appositi diagrammi sotto forma di parallelogrammi orientati in modo opposto agli obiettivi (Figura 12).

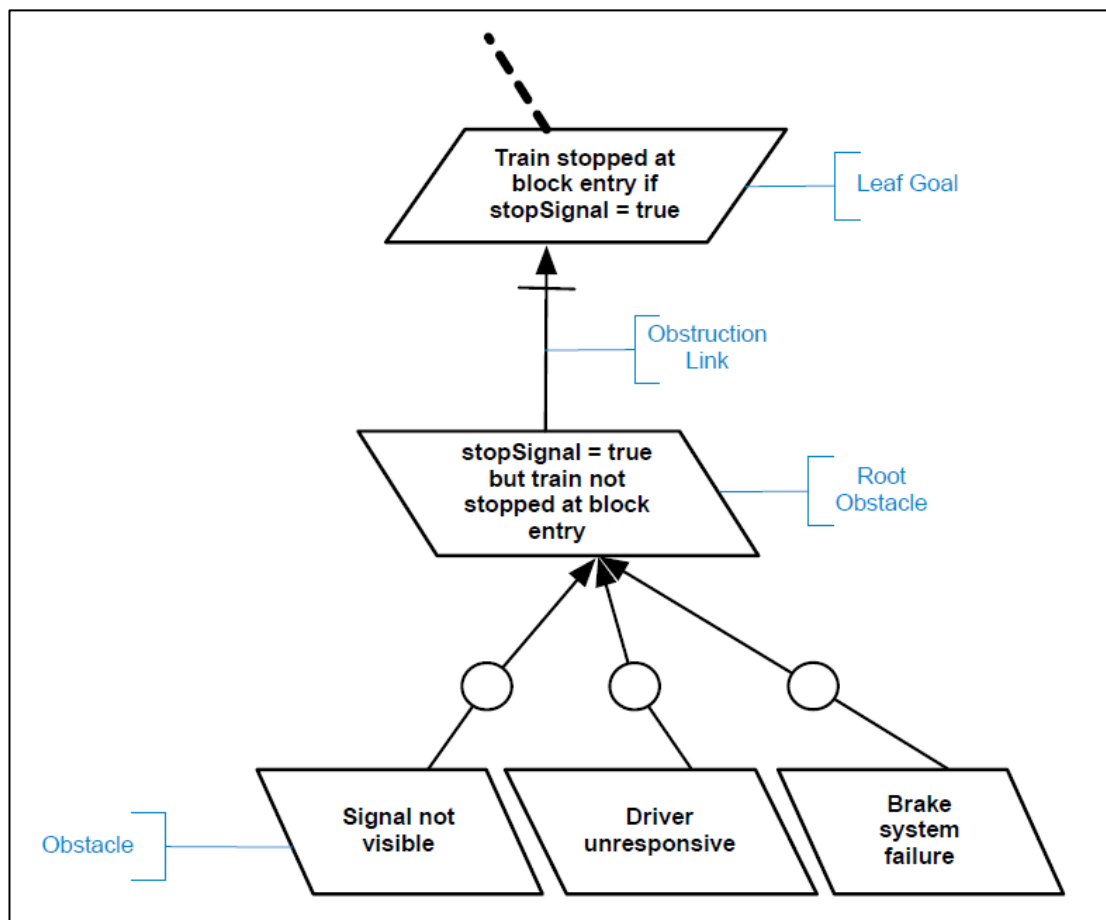


Figura 12: Diagramma di un ostacolo

L'obiettivo dell'esempio in Figura 12 riporta che se il segnale di stop è acceso, allora il treno dovrebbe fermarsi. L'ovvio ostacolo a questo obiettivo è che il segnale sia acceso, ma il treno continui la sua corsa. Un ostacolo diretto ad un obiettivo è detto ostacolo radice (*root obstacle*). L'ostacolo radice dell'esempio è decomposto in tre fattori che possono contribuire all'ostacolo: il segnale di stop potrebbe non essere visibile, il macchinista del treno potrebbe non essere in condizione di azionare il freno, e lo stesso freno potrebbe aver subito un'avaria. Anche gli ostacoli, così come gli obiettivi, si possono raffinare attraverso AND ed OR.

L'identificazione e la decomposizione degli ostacoli segue gli stessi principi dell'analisi degli alberi dei guasti (*Fault Tree Analysis* o *FTA*). L'utilizzo degli alberi dei guasti nell'ambito della qualificazione delle tecnologie è già noto, e sono già esistenti delle linee guida per la loro applicazione in contesti industriali [19].

Il vantaggio principale che offre Modus è collegare gli alberi dei difetti presenti a bassi livelli di astrazione ad obiettivi di livello più alto. Da un lato ciò fornisce una misura di completezza per l'insieme degli alberi costruiti: l'analista può esaminare esaustivamente l'insieme degli obiettivi foglia nel modello e decidere quali obiettivi necessitino di un'analisi degli alberi dei guasti. D'altra parte, collegare gli obiettivi agli alberi dei guasti può presentare considerevoli benefici in fase di elicitazione. Nello specifico, per molte tipologie di obiettivi, è più semplice raccogliere evidenze ed esprimere opinioni riguardo la probabilità di ostacoli che possano negare un obiettivo, piuttosto che la probabilità di soddisfacimento di un obiettivo stesso. Questo aspetto di Modus propone una soluzione alla problematica della mancata specifica dei metodi e meccanismi di fallimento (Sezione 1.3.3).

2.5 Pianificazione della raccolta delle evidenze

Decomposizione di obiettivi ed identificazione di ostacoli forniscono una base per la pianificazione della raccolta delle evidenze. Nello specifico, sarebbe opportuno raccogliere le evidenze in modo tale da facilitare agli esperti il compito di rispondere a domande del tipo:

- “Quanto è probabile che un obiettivo foglia sia soddisfatto?”
- “Quanto è probabile che si presenti l’occorenza di un ostacolo foglia?”

Le evidenze possono consistere di vari artefatti, e saranno collegate agli obiettivi e agli ostacoli foglia. La Figura 13 mostra la rappresentazione delle evidenze in un diagramma sotto forma di ovali.

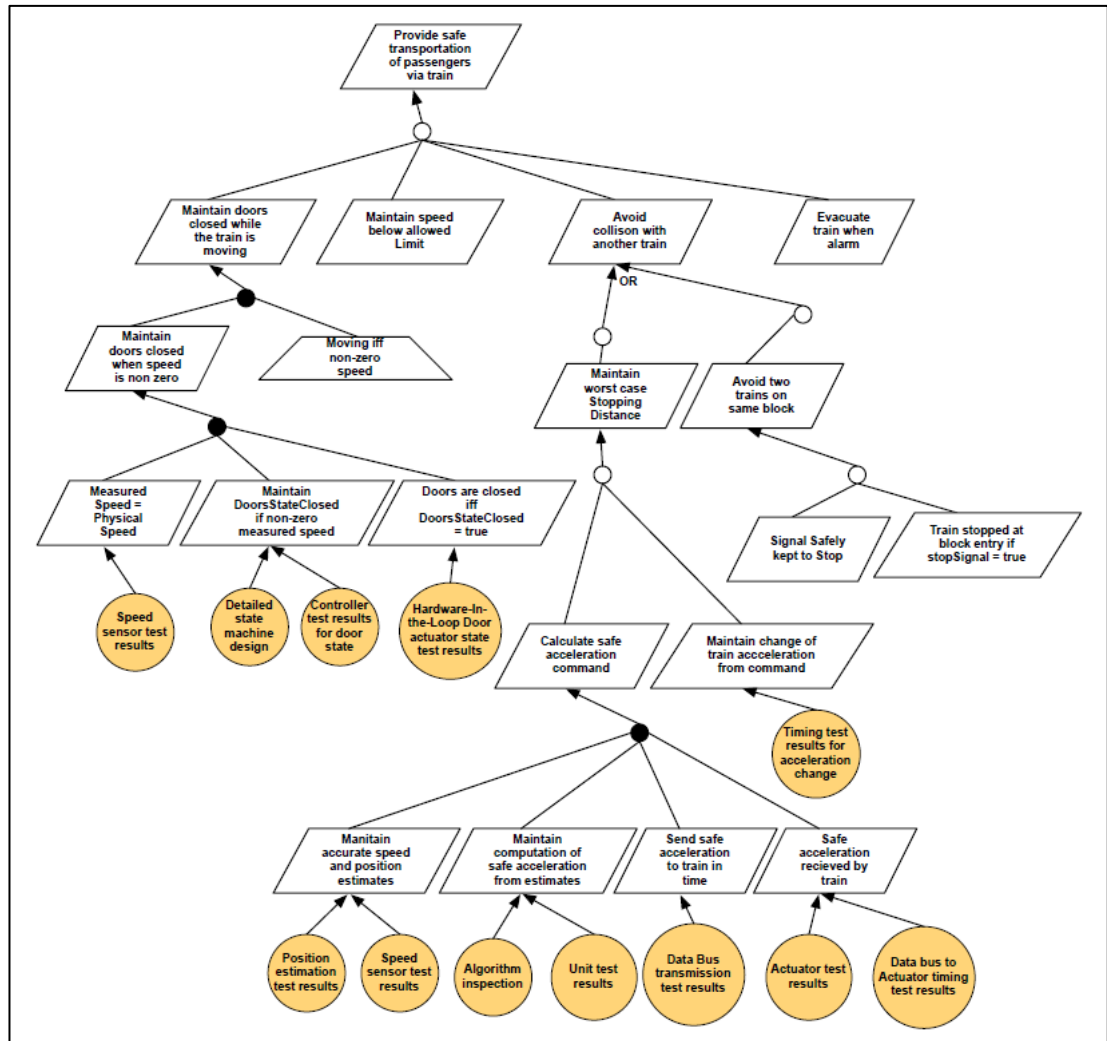


Figura 13: Supporto degli obiettivi foglia attraverso evidenze

Aver sviluppato un modello degli obiettivi prima di pianificare la raccolta delle evidenze, rende quest'ultimo processo più efficace, ed evita la raccolta di informazioni inutili o irrilevanti, consentendo un risparmio di effort, tempi e costi. Questo aspetto di Modus mira a risolvere la problematica correlata alla tardiva raccolta delle evidenze (Sezione 1.3.2).

Per facilitare la pianificazione della raccolta delle evidenze, Modus definisce una tassonomia degli artefatti che possono essere utilizzati a supporto degli obiettivi (Figura 14).

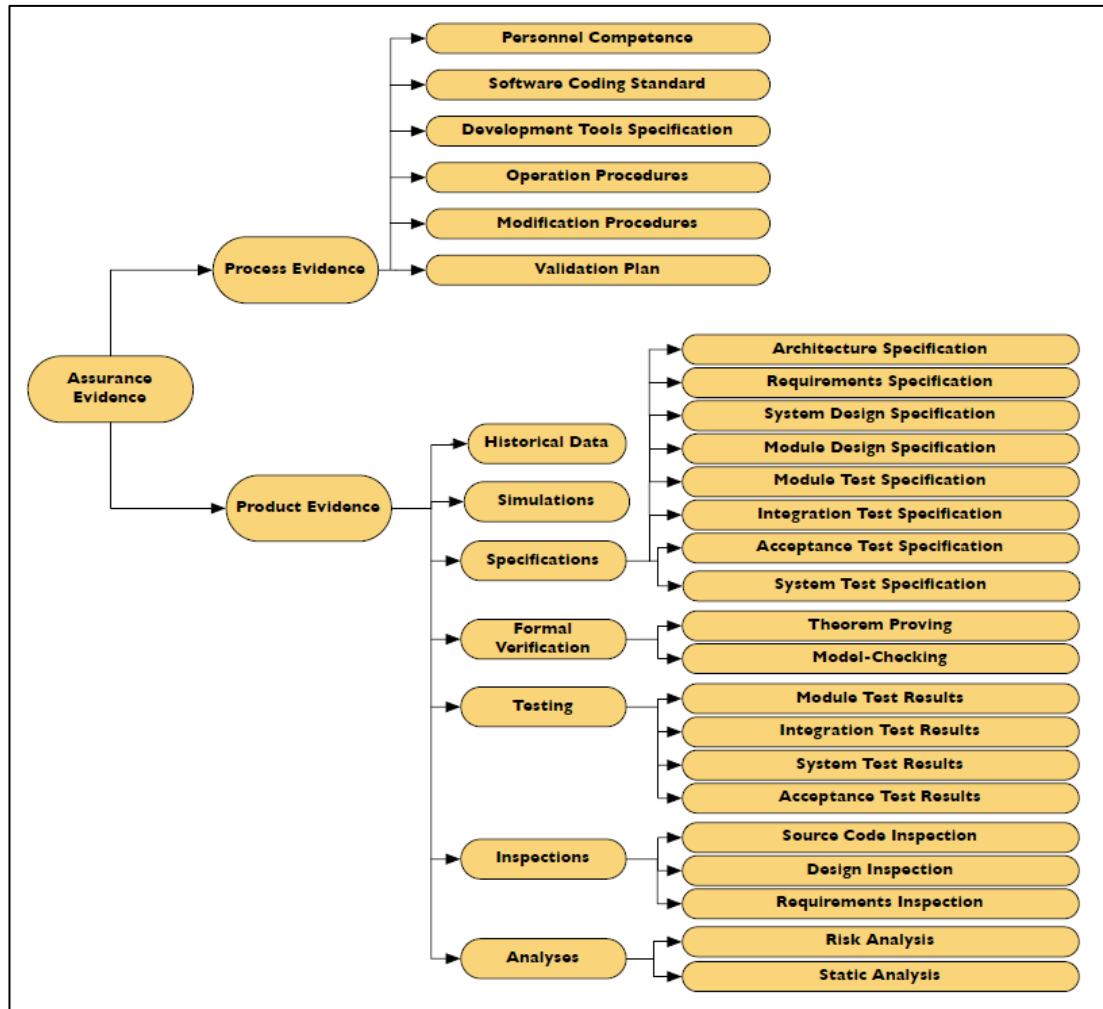


Figura 14: Classificazione delle evidenze

Nonostante sia una risorsa di valore per gli analisti durante la pianificazione della raccolta delle evidenze, la classificazione non deve essere considerata come completamente esaustiva: esistono infatti diverse evidenze *domain-specific* (e per questo non incluse nella tassonomia) che devono essere considerate durante il processo.

Una volta raccolte, le evidenze collegate agli obiettivi o ostacoli foglia forniranno agli esperti le informazioni necessarie alla formazione di un'opinione in merito alla probabilità di soddisfacimento di un obiettivo o di avvenimento di un ostacolo. Le probabilità saranno in seguito propagate nel modello, per valutare se gli obiettivi strategici di più alto livello siano adeguatamente soddisfatti (Sezione 2.7).

Come già osservato nella Sezione 2.1, Modus non ha l'obiettivo di fornire linee guida per la raccolta in sé delle evidenze, che deve essere decisa in accordo a fattori dipendenti dal contesto applicativo, così come standard e normative in atto nel sistema da sviluppare.

2.6 Elicitazione delle quantità dagli esperti

Per essere in grado di giungere a conclusioni significative sul grado di soddisfazione degli obiettivi strategici attraverso la propagazione automatica dei dati forniti dagli esperti, è fondamentale assicurare di disporre di opinioni quanto più accurate possibile. Modus utilizza un approccio probabilistico per garantire la quantificazione dei risultati proposti. L'utilizzo delle probabilità presenta due vantaggi principali: in primo luogo, le probabilità risultano molto intuitive ai professionisti, e spesso vi è ben poca confusione riguardo la loro interpretazione [22]. Un altro vantaggio, che deriva dalla popolarità dei modelli probabilistici, consiste nel fatto che le prassi per l'elicitazione delle probabilità sono state studiate a fondo ed esiste pertanto un'ampia letteratura in merito i cui risultati possono essere efficacemente utilizzati [22] [23].

In casi in cui, come la qualificazione di nuove tecnologie, non si può fare affidamento su dati passati (come tassi di guasto per alcuni componenti), è necessario che gli esperti di dominio forniscano delle stime accurate per alcune quantità di interesse. Per componenti hardware e meccanici che sono stati utilizzati in tecnologie precedenti, spesso esistono sufficienti dati storici riguardo la sicurezza funzionale (in particolar modo per quanto concerne l'affidabilità), prevalentemente in forma di distribuzioni di probabilità. La dipendenza dalla conoscenza degli esperti in questi casi è significativamente minore, ma comunque necessaria per interpretare ed adattare i dati alla nuova tecnologia.

2.6.1 Descrizione delle quantità che gli esperti devono specificare

Modus definisce tre tipi di probabilità i cui valori devono essere forniti dagli esperti:

- **Probabilità di soddisfacimento di un obiettivo foglia:** dato un obiettivo foglia G , supportato delle evidenze $E_1, E_2, \dots E_n$, si pone agli esperti la seguente domanda: “Qual è la probabilità che $E_1, E_2, \dots E_n$ soddisfino i requisiti di G ?”. Questo tipo di analisi viene condotta quando non si manifesta la necessità di procedere con l’analisi degli ostacoli.
- **Probabilità di avvenimento di un ostacolo foglia:** dato un ostacolo foglia O , supportato delle evidenze $E_1, E_2, \dots E_n$, si pone agli esperti la seguente domanda: “Qual è la probabilità che avvenga O date $E_1, E_2, \dots E_n$?”. Questo tipo di analisi è simile a quella condotta per gli obiettivi. Vi è però una differenza sostanziale nel significato della quantità elicitata: gli ostacoli negano il soddisfacimento degli obiettivi e pertanto la loro probabilità di occorrenza va intesa in senso complementare.
- **Probabilità di completezza di una decomposizione:** come descritto nella Sezione 2.3.1, la decomposizione di obiettivi ed ostacoli può essere parziale. Dato un obiettivo G , ed i suoi sotto-obiettivi $G_1, G_2, \dots G_n$, si pongono agli esperti la seguenti domande:
 - Q_1 (**AND parziali**): “Qual è la probabilità G fallisca anche quando $G_1, G_2, \dots G_n$ sono soddisfatti?”
 - Q_2 (**OR parziali**): “Qual è la probabilità G sia soddisfatto anche quando nessuno tra $G_1, G_2, \dots G_n$ è soddisfatto?”Analogamente, dato un ostacolo O ed i suoi sotto-ostacoli $O_1, O_2, \dots O_n$, si pongono agli esperti la seguenti domande:
 - Q'_1 (**AND parziali**): “Qual è la probabilità O non si verifichi anche quando $O_1, O_2, \dots O_n$ si sono verificati?”
 - Q'_2 (**OR parziali**): “Qual è la probabilità O si verifichi anche quando nessuno tra $O_1, O_2, \dots O_n$ si è verificato?”

In contesti di valutazione del rischio, come quello in cui Modus intende operare, è ragionevole mantenere un approccio conservativo e trattare le decomposizioni parziali per gli OR sugli obiettivi e le decomposizioni parziali degli AND sugli ostacoli come fossero complete. Un OR parziale per gli obiettivi esprime il concetto secondo cui delle “alternative” per il soddisfacimento dello stesso potrebbero essere aggiunte in futuro. Per ragioni legate alla gestione del rischio, si potrebbe supporre che queste alternative non vengano mai aggiunte. Per gli ostacoli accade invece l’opposto: un OR parziale implica che potrebbero esistere modalità alternative secondo cui potrebbe avvenire un determinato ostacolo delle quali non si è ancora a conoscenza. È importante che sia considerato questo grado di incertezza in fase di valutazione del rischio. Al contrario invece, gli AND parziali per gli ostacoli possono essere ignorati, in quanto completare con ulteriori eventi la decomposizione può solo diminuire la probabilità di avvenimento dell’ostacolo ed aumentare pertanto il soddisfacimento degli obiettivi. Gli AND parziali per gli obiettivi devono invece essere considerati, in quanto rappresentano l’eventualità che di un determinato obiettivo non si conoscano in primo luogo tutte le eventualità che simultaneamente ne devono determinare il soddisfacimento. Su queste basi, le domande Q_2 e Q'_1 possono essere ignorate in fase di elicitazione. Q_1 e Q'_2 devono però essere considerate in quanto sono volte a considerare i rischi nascosti causati dalle incompletezze delle decomposizioni.

2.6.2 Processo di elicitazione

La Tabella 2 riporta una visione d’insieme della procedura di elicitazione delle opinioni degli esperti. Il processo consiste di tre fasi, di cui la fase centrale rappresenta l’effettivo step di elicitazione.

Fasi	Sottofasi	Descrizione

Pre-operazione: preparazione	Preparazione dei soggetti	Nominare, dare istruzioni, preparare e motivare gli esperti
	Preparazione del processo	Definire tempi, costi, obiettivi dell'elicitazione e domande
	Preparazione degli oggetti	Preparare il materiale
Operazione: elicitazione	Esecuzione	Chiedere le domande in maniera corretta
	Raccolta	Far rispondere gli esperti
	Verifica	Controllare le consistenze tra le risposte
Post-operazione: analisi delle probabilità	Aggiustamento	Modificare le stime con operazioni di <i>scaling</i> e <i>smoothing</i>
	Aggregazione	Aggregare i risultati di diversi esperti
	Creazione delle probabilità	Tradurre le stime in distribuzioni di probabilità

Tabella 2: Visione d'insieme del processo di elicitazione

- **Pre-operazione: preparazione:** insieme delle attività da svolgere in preparazione dell'elicitazione. Nello specifico, occorre scegliere gli esperti, motivarli illustrandogli gli scopi dell'elicitazione, dargli istruzioni mostrando esempi, procedure ed errori comuni. La selezione degli esperti è guidata da due criteri: il primo consiste nel ruolo che gli esperti hanno nello sviluppo e nel processo di valutazione. Gli esperti devono comunque avere accesso alle informazioni che dovrebbero stimare. Il secondo criterio per la selezione degli esperti è il loro livello di esperienza, che deve essere sufficiente per poter completare efficacemente l'attività. È molto importante, inoltre, che gli esperti siano ben motivati in modo che essi forniscano risposte accurate. Durante la fase di preparazione, vengono stilate descrizioni non ambigue per le quantità da elicitare, al fine di poter creare gli strumenti che saranno

utilizzati nella fase successiva. Questi consistono di questionari a cui gli esperti devono rispondere, e procedure di interviste per guidare l'intervistatore durante le domande. Lo scopo della fase è quindi rendere le interviste più sistematiche e consistenti in modo da poter disporre, ad attività ultimata, di stime più affidabili.

- **Operazione: elicitazione:** pianificazione di un'intervista con un esperto. È sempre preferibile che le interviste siano faccia a faccia, in quanto gli esperti sono più motivati e l'intervistatore mantiene un controllo maggiore sul processo. Comunque, a seconda di quante domande debbano essere chieste agli esperti, è possibile procedere con interviste telefoniche. Durante l'intervista, l'intervistatore guida gli esperti nello svolgimento dell'attività, utilizzando una procedura ben definita. È fondamentale monitorare nel processo l'eventuale presenza di errori che potrebbero influenzare le risposte degli esperti. Per questo motivo, vengono svolte due attività:
 - **Informare gli esperti della potenziale presenza di errori:** studi sperimentali riguardo l'elicitazione delle probabilità hanno infatti dimostrato che un sostanziale miglioramento della qualità della valutazione può essere ottenuto preparando gli esperti all'elicitazione [23]. È pertanto necessario che gli esperti siano a conoscenza dei pregiudizi che di solito hanno (Sezione 2.6.4), e nello specifico definizione e cause di essi. A tal fine, durante le interviste si presenta il concetto generale di errore di elicitazione ed una lista degli errori più comuni.
 - **Monitorare l'elicitazione per rilevare l'eventuale presenza di errori:** durante l'elicitazione, l'intervistatore osserva il linguaggio del corpo espresso dall'esperto ed il modo con cui egli esprime le opinioni. Se alcuni segnali sono indice di possibile presenza di pregiudizi, l'intervistatore dovrebbe agire di conseguenza. Un esempio è costituito dalle frasi di tipo "abbiamo avuto un caso in cui..." che potrebbe portare ad un possibile *outlier* nella distribuzione

di probabilità che sarà elicitata. Ogni volta che l'esperto motiva in questo modo le sue risposte, l'intervistatore dovrebbe indagare sull'effettiva rappresentatività del caso addotto a motivazione.

Una volta completate tutte le interviste, avviene la comparazione delle risposte degli esperti. Nel caso si osservino differenze significative, occorre che queste siano investigate accuratamente. È possibile motivare la registrazione di una risposta sensibilmente differente dalle altre, con il fatto che l'esperto che l'ha fornita potrebbe aver avuto un'esperienza particolare. In questo caso, bisogna valutare quanto questa esperienza sia significativa. Al termine di quest'attività si dispone, per ogni domanda del questionario, di un insieme di risposte degli esperti rappresentate quantitativamente. Bisogna inoltre verificare i dati forniti dagli esperti, chiedendo loro conferma riguardo i risultati derivanti dall'aggregazione delle loro risposte e controllando che la somma delle probabilità di tutti i possibili esiti di un evento sia pari ad 1.

- **Post-operazione: analisi delle probabilità:** studio dei valori elicitati e creazione delle distribuzioni. I valori degli esperti sono prima corretti nel caso in cui si presentino errori, e successivamente aggregati in distribuzioni di probabilità.

2.6.3 Elicitazione di distribuzioni sconosciute

In assenza di informazioni riguardo le caratteristiche della distribuzione di probabilità da elicitare, è preferibile utilizzare una distribuzione triangolare al posto di distribuzioni parametriche: ciò è motivato dal fatto che i parametri della distribuzione triangolare sono i più intuitivi per gli esperti, e le domande che ne chiedono il valore sono le più facili a cui essi possano rispondere [24]. Funzione di densità e funzione di distribuzione cumulativa della distribuzione triangolare sono riportate in Figura 15 e Figura 16.

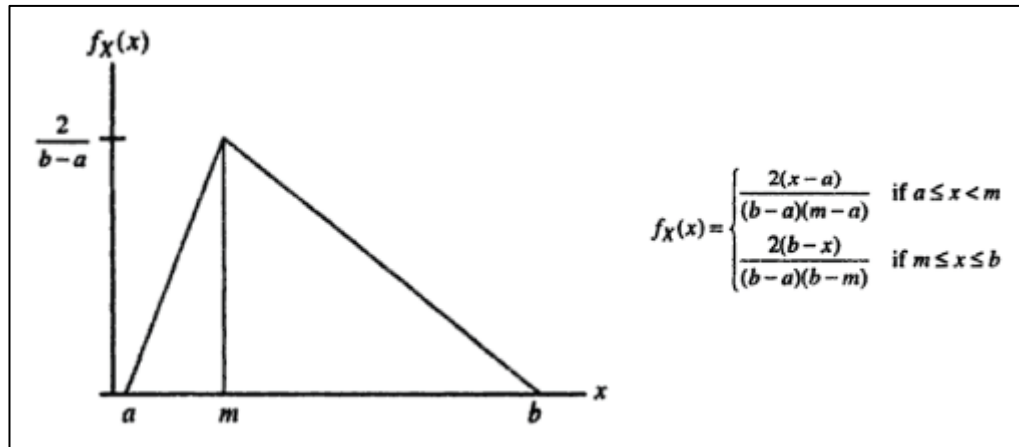


Figura 15: Funzione di densità della distribuzione triangolare

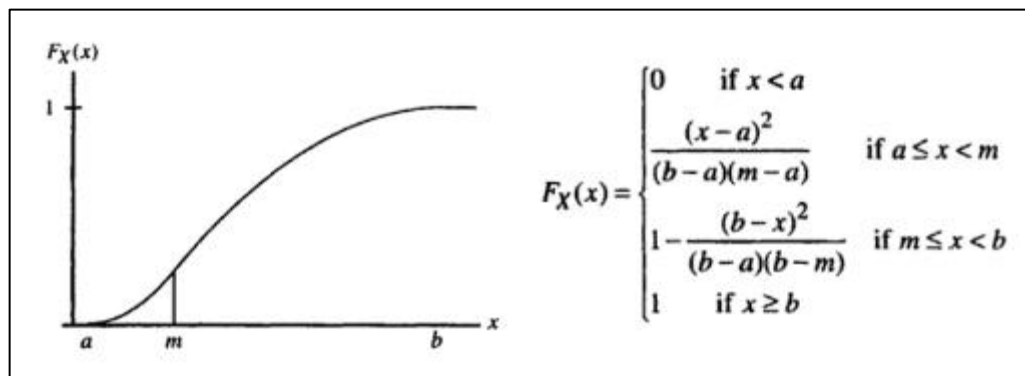


Figura 16: Funzione di distribuzione cumulativa della distribuzione triangolare

Per elicitarre una distribuzione triangolare sono richiesti tre parametri: il valore atteso m , il valore minimo b ed il valore massimo a . Esistono studi in letteratura [25] [26] che suggeriscono all'intervistatore di chiedere prima il valore massimo e lasciar focalizzare l'esperto su quello, al fine di prevenire eventuali effetti di consistenza forzata tra le risposte (Sezione 2.6.4).

Dal momento che si tende in genere a sottistimare il valore minimo e sovrastimare il valore massimo, si opta in genere per un allargamento dell'intervallo di confidenza della distribuzione [27], in genere utilizzando un approccio secondo cui gli estremi della distribuzione sono allargati distribuendo equamente una percentuale fissata della distribuzione prima del valore minimo e dopo il valore massimo. La Figura 17 presenta un esempio in cui l'esperto ha fornito i valori $a = 0.85, m = 0.90, b =$

0.96. Distribuendo il 5% dell'area della distribuzione attorno agli estremi, si ottiene una distribuzione triangolare identificata dai valori $a = 0.82548$, $m = 0.90$, $b = 0.9863$.

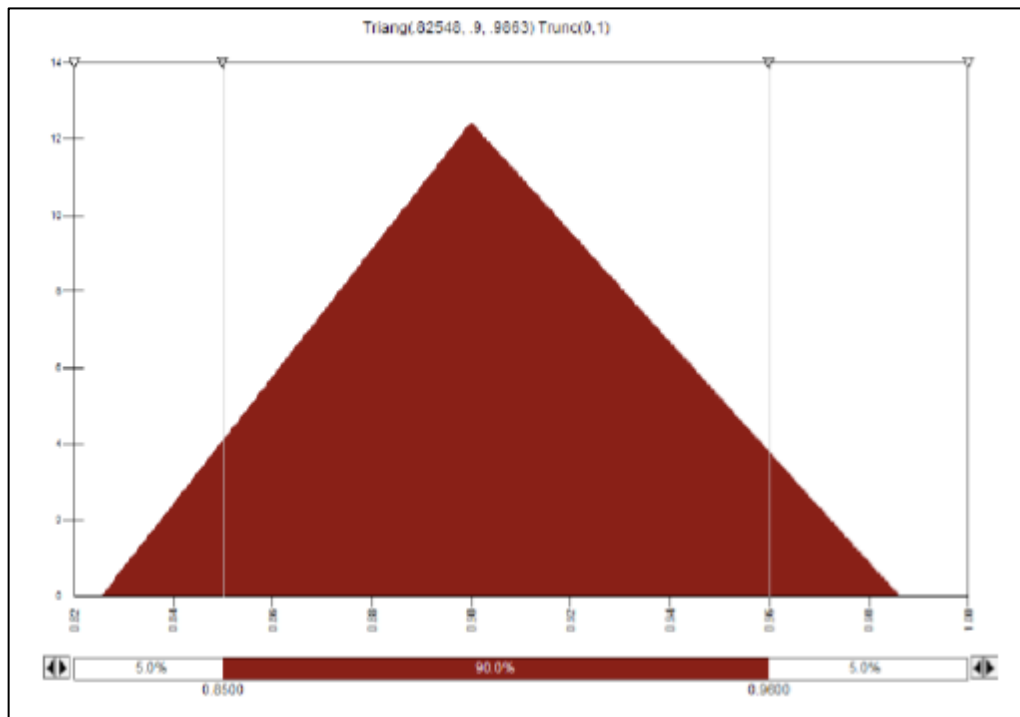


Figura 17: Esempio di distribuzione triangolare con gli intervalli di confidenza estesi

La percentuale esatta utilizzata per l'allargamento degli intervalli di confidenza può essere decisa solo da fattori dipendenti dal contesto applicativo, e Modus non specifica un intervallo di valori consigliati, limitandosi a raccomandare un'estensione di una percentuale fissata.

2.6.4 Gestione degli errori in fase di elicitazione

Gli esperti possono consciamente o inconsciamente polarizzare i risultati dell'elicitazione. Errori in questo senso possono riferirsi a incomprensioni circa le domande, analisi delle informazioni rilevanti per le risposte ed effettiva comunicazione di esse. All'inizio dell'intervista, gli esperti devono essere informati

riguardo la potenziale presenza di questi errori. A tal proposito, occorre introdurre le definizioni dei diversi tipi di errore che possono introdurre distorsioni nei dati [28]:

- **Inconsistenza:** a causa di stanchezza o confusione, gli esperti potrebbero modificare le loro risposte.
- **Consistenza forzata:** nel caso in cui una domanda sia correlata ad un'altra, gli esperti potrebbero tendere a forzare risposte consistenti anche nel caso in cui abbia a disposizione nuove informazioni.
- **Sottostima dell'incertezza:** gli esperti potrebbero sottostimare l'incertezza delle loro risposte in quanto si sentono in dovere di conoscere perfettamente un evento e la sua probabilità di occorrenza.
- **Motivazionale:** gli esperti potrebbero tendere ad accontentare l'intervistatore. In questo modo la risposta alle domande dipende fortemente dalla loro formulazione, piuttosto che dal loro contenuto
- **Accordo di gruppo:** nel caso in cui diversi esperti rispondano alla stessa domanda, essi potrebbero involontariamente fornire le stesse risposte per evitare disaccordi
- **Interessi personali:** gli esperti potrebbero fornire risposte in accordo con i loro interessi personali. Ad esempio se un esperto è incaricato di assicurare l'affidabilità di un componente, potrebbe sovrastimarne le effettive prestazioni.
- **Interessi generali:** gli esperti potrebbero tendere a fornire delle risposte che accontentino gli stakeholder. Ad esempio, un esperto potrebbe sovrastimare l'affidabilità del sistema in modo che gli stakeholder siano soddisfatti di essa.
- **Modalità di risposta:** mappare un concetto in una modalità di risposta è un compito difficile. Le risposte degli esperti potrebbero cambiare in accordo ai diversi tipi di modalità di risposta. Il processo analitico gerarchico (*Analytic Hierarchy Process* o *AHP*) [29] è un esempio di meccanismo utilizzato per tradurre concetti soggettivi in dati quantitativi che definisce le modalità di risposta come livelli di una scala ordinale.

Dato lo spesso volte significativo impatto negativo degli errori nelle stime di soddisfacimento degli obiettivi, è opportuno definire delle procedure per gestirlo correttamente. Nello specifico è possibile:

- **Progettare** la fase di elicitazione in modo da evitare in anticipo alcuni errori.
- **Istruire** gli esperti riguardo i possibili tipi di errori.
- **Monitorare durante l'intervista** l'occorrenza di errori, migliorando la progettazione della fase se necessario.
- **Controllare durante l'analisi dei dati** l'occorrenza di eventuali errori e correggere i risultati.

2.7 Analisi del soddisfacimento degli obiettivi strategici

Una volta specificate le quantità descritte nella Sezione 2.6.1, i valori devono essere propagati per computare una distribuzione di probabilità circa il soddisfacimento di ognuno degli obiettivi strategici. Si presenta di seguito un algoritmo per la propagazione dei valori puntuali di probabilità (Sezione 2.7.1) ed una modalità di propagazione delle distribuzioni di probabilità attraverso simulazioni Monte Carlo (Sezione 2.7.2).

2.7.1 Propagazione dei valori puntuali di probabilità

L'algoritmo di propagazione dei valori puntuali è un adattamento dell'algoritmo di propagazione degli obiettivi riportato in [30]. La Figura 18 riporta le 5 regole alla base dell'algoritmo: le regole 1-4 si riferiscono alla propagazione dei valori tra obiettivi, e sono analoghe a quelle per la propagazione dei valori tra ostacoli, che

non verranno ripetute. La regola 5 specifica la propagazione da ostacoli a goal foglia.

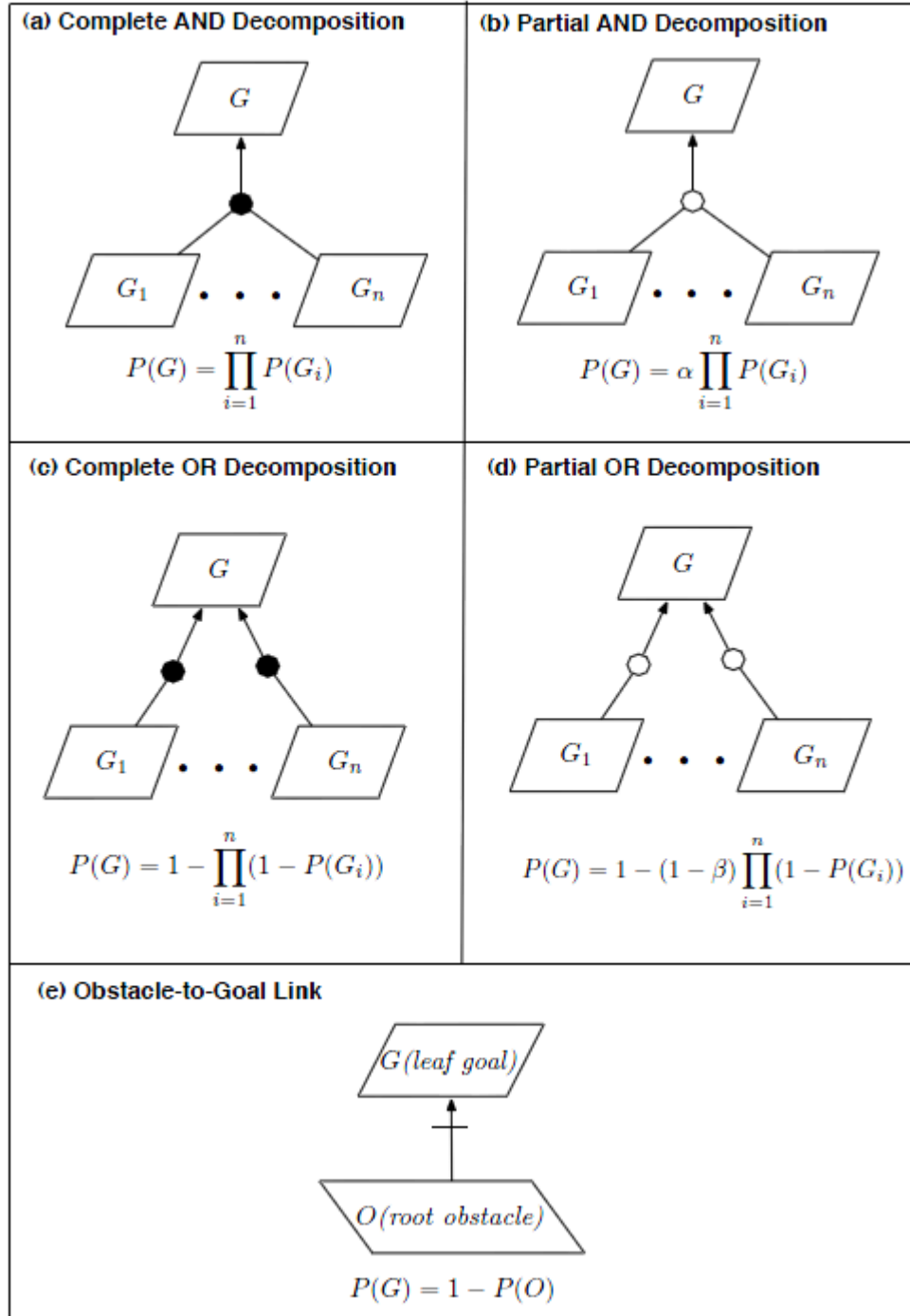


Figura 18: Regole per la propagazione dei valori puntuali di probabilità

- **Regola 1 (a):** regola di propagazione per la decomposizione di un obiettivo secondo un AND completo. Una decomposizione completa implica che il

soddisfacimento di tutti i sotto-obiettivi G_1, G_2, \dots, G_n , è sufficiente per il soddisfacimento dell'obiettivo padre G . Assumendo l'indipendenza di G_1, G_2, \dots, G_n la probabilità di soddisfacimento di G è il prodotto delle probabilità di soddisfacimento di G_1, G_2, \dots, G_n .

- **Regola 2 (b):** regola di propagazione per la decomposizione di un obiettivo secondo un AND parziale. In questo caso la decomposizione deve considerare l'incompletezza della decomposizione, il cui impatto è stimato dal valore $\alpha \in (0, 1)$. Il valore $(1 - \alpha)$ deve essere elicitato dall'esperto attraverso la domanda Q_1 della Sezione 2.6.1. In altre parole, il valore di α è la risposta alla domanda: "Qual è la probabilità che ci siano obiettivi sconosciuti o non specificati da essere soddisfatti per soddisfare G ?". Tuttavia, è più naturale elicitare il valore di $(1 - \alpha)$, piuttosto che quello di α . Per i motivi riportati nella Sezione 2.6.1, la regola corrispondente per la decomposizione di ostacoli non viene attuata, riconducendo il caso a quello di una decomposizione completa.
- **Regola 3 (c):** regola di propagazione per la decomposizione di un obiettivo secondo un OR completo. G_1, G_2, \dots, G_n possono essere considerati ognuno come un'alternativa per il soddisfacimento di G . In accordo con la teoria delle probabilità, la probabilità di soddisfare G è il complemento ad 1 della probabilità che non sia soddisfatta alcuna alternativa.
- **Regola 4 (d):** regola di propagazione per la decomposizione di un obiettivo secondo un OR parziale. Anche in questo caso la decomposizione deve considerare l'incompletezza della decomposizione, il cui impatto è stimato dal valore $\beta \in (0, 1)$. Il valore β deve essere elicitato dall'esperto attraverso la domanda Q_2 della Sezione 2.6.1. Per i motivi riportati nella Sezione 2.6.1, questa regola è attuata solo per la decomposizione di ostacoli (per cui è prevista la domanda Q'_2), mentre per la decomposizione di obiettivi si riconduce il caso a quello di una decomposizione completa.

- **Regola 5 (e):** regola per la propagazione dell'ostruzione di un ostacolo ad un obiettivo foglia. Il valore propagato in questo caso è semplicemente il complemento ad 1 della probabilità dell'ostacolo che ostruisce l'obiettivo.

Date queste regole, la propagazione può essere computata con l'algoritmo riportato in Tabella 3.

- Contrassegna tutti gli obiettivi foglia come "esaminati";
- **while** (la radice dell'albero non è "esaminata")
 - for (ogni goal o ostacolo X per cui tutti i figli sono "esaminati")
 - Applica la regola di propagazione della Figura 18;
 - Contrassegna X come "esaminato";
 - **end for**;
- **end while**;
- **output** valore della radice dell'albero;

Tabella 3: Algoritmo di propagazione dei valori puntuali

L'assunzione principale effettuata durante la propagazione dei valori è quella secondo cui i sotto-obiettivi di ogni obiettivo sono indipendenti tra loro. Nel caso generale, dato un insieme completo di sotto-obiettivi G_1, G_2, \dots, G_n per un obiettivo padre G , la probabilità $P(G)$ di soddisfacimento di G è espressa da:

$$P(G) = P(G_1 | G_2, G_3 \dots G_n) * P(G_2 | G_3, G_4 \dots G_n) * \dots * P(G_{n-1} | G_n) * P(G_n)$$

Dove $P(X | Y)$ indica la probabilità condizionata di X dato Y .

Solamente quando G_1, G_2, \dots, G_n sono indipendenti, il valore di $P(G)$ è pari a $\prod_{i=1}^n P(G_i)$. Quest'assunzione è comunque comune nella teoria relativa alla propagazione degli obiettivi [30].

Accade inoltre spesso, in merito alla definizione di ostacoli, che ci siano cause di fallimento comuni come mancanza di energia elettrica, allagamenti o errori umani: ciò comporta che un ostacolo possa ostruire più obiettivi simultaneamente. Sebbene esistano alcuni approcci per lo studio di cause di fallimento comuni [31],

Modus segue l'approccio standard dell'FTA, secondo cui ostacoli comuni sono riportati per ogni loro occorrenza in copie multiple all'interno dei modelli di analisi.

2.7.2 Propagazione delle distribuzioni di probabilità

L'algoritmo riportato nella Tabella 3 è sufficiente a propagare i dati se i dati forniti dagli esperti sono valori puntuali di probabilità. Tuttavia vi è spesso incertezza riguardo l'esatta formulazione dei valori, ragion per cui è necessario avere una distribuzione di probabilità per ogni obiettivo strategico. La computazione di tali probabilità avviene con l'utilizzo del metodo Monte Carlo [24] [32], una tecnica basata su ripetuti campionamenti casuali da un insieme di variabili di input al fine di ottenere un'approssimazione probabilistica di un insieme di variabili di output.

Nel caso di Modus, le variabili \bar{x}_i di input sono le quantità specificate dagli esperti, e la variabile \bar{y} di output è il grado di soddisfazione di un obiettivo strategico, come riportato in Figura 19.

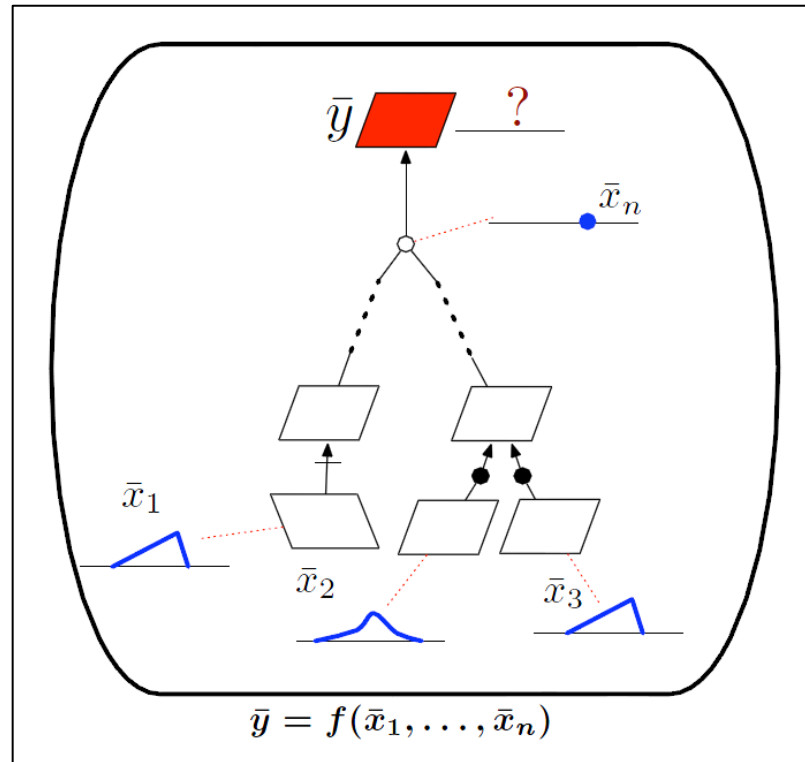


Figura 19: Propagazione dei valori di probabilità nel modello in termini di variabili di input e variabili di output

La Figura 20 riporta invece l'algoritmo Monte Carlo utilizzato per computare la distribuzione di probabilità indice del grado di soddisfacimento di un obiettivo.

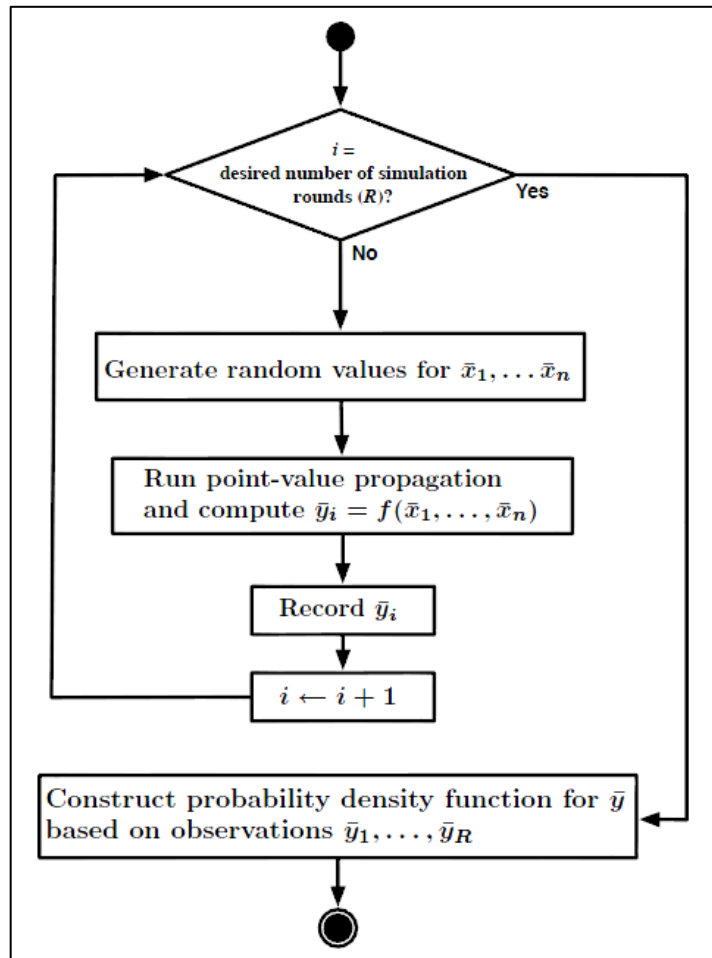


Figura 20: Algoritmo Monte Carlo di propagazione delle distribuzioni di probabilità

Per eseguire l'algoritmo, occorre impostare il numero R di iterazioni dell'algoritmo. In ogni iterazione, si generano variabili casuali di input in accordo alla distribuzione elicitata dall'esperto. Quindi, si esegue l'algoritmo di propagazione riportato nella Tabella 3 della Sezione 2.7.1 computando il valore risultante di \bar{y}_i . Dopo aver eseguito l'algoritmo per R iterazioni, si sviluppa una curva approssimata di probabilità per il valore di \bar{y} computando la frequenza dei valori osservati che cadono all'interno del range di valori tra il minimo ed il massimo osservati tra gli \bar{y}_i .

Un esempio relativo alla computazione della distribuzione di probabilità per un obiettivo strategico è riportato in Figura 21, dove viene ripreso l'esempio della Figura 10 della Sezione 2.3.1.

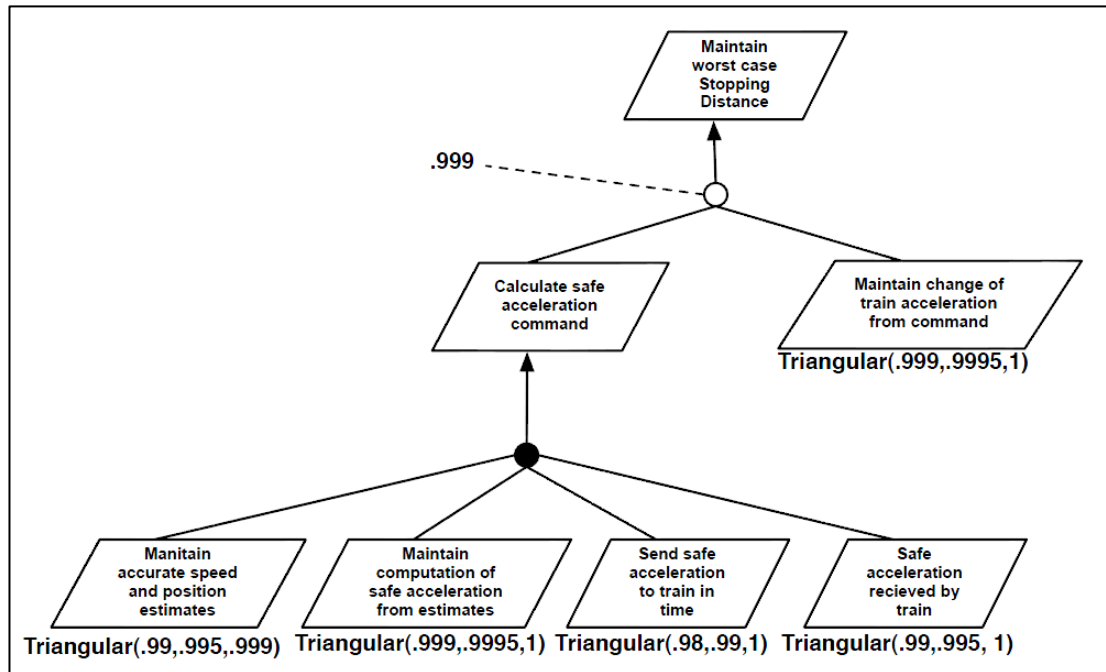


Figura 21: Esempio di un modello degli obiettivi con le distribuzioni di probabilità degli obiettivi foglia

Ad ogni obiettivo foglia sono state assegnate delle distribuzioni di probabilità fittizie, inoltre è riportato anche un valore puntuale associato alla decomposizione parziale dell'obiettivo padre. Una simulazione Monte Carlo con $R = 3000$ iterazioni per l'obiettivo padre conduce alla definizione della distribuzione di probabilità riportata in Figura 22.

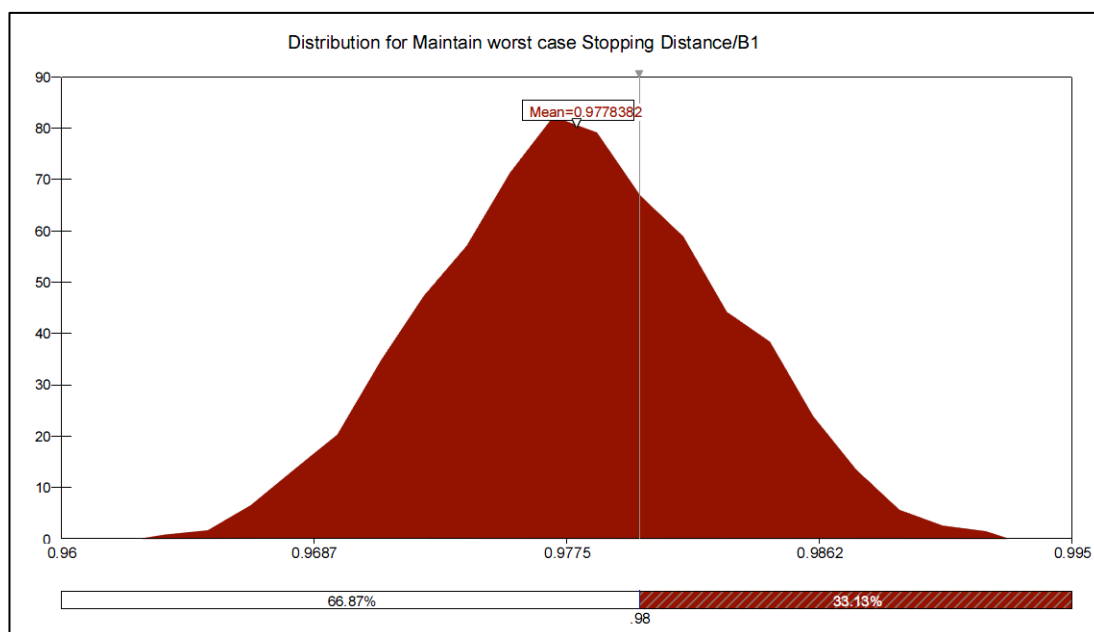


Figura 22: Esempio di distribuzione di probabilità risultante dall'esecuzione di una simulazione Monte Carlo per un obiettivo strategico

La curva mostra non solo la probabilità media di soddisfacimento dell'obiettivo, ma anche il livello di incertezza associato con il soddisfacimento. Ad esempio, se la probabilità di soddisfacimento desiderata (*target*) per l'obiettivo radice è del 98%, la probabilità che la tecnologia soddisfi quest'obiettivo, a fronte dello studio condotto, è stimata in una quantità prossima al 33%. Questo basso valore della stima può essere causato da più fattori:

- Le evidenze a supporto del soddisfacimento degli obiettivi foglia sono scarse o inadeguate;
- Gli esperti sono stati troppo conservativi nelle loro stime a causa di pregiudizi riguardanti alcune precedenti omissioni di parametri riguardo l'ambiente operativo della tecnologia (Sezione 1.3.6);
- Le misure prese per assicurare il soddisfacimento dell'obiettivo radice sono inadeguate (ad esempio non c'è replicazione dei componenti più critici del sistema), e pertanto il modello deve essere rielaborato.

In ognuno di questi casi, è importante identificare quali quantità del modello abbiano un impatto significativo sul soddisfacimento degli obiettivi di livello più

alto. A tal proposito risulta utile un'analisi di sensibilità (*sensitivity analysis*) riportata nella Sezione 2.8.

2.8 Miglioramento di obiettivi, evidenze e stime degli esperti

Per migliorare il grado di soddisfacimento degli obiettivi strategici è importante dapprima identificare i fattori che contribuiscono significativamente a questi obiettivi. Quest'operazione è nota come analisi di sensibilità, e consente agli analisti di focalizzare la loro attenzione sul miglioramento dei fattori che impattano maggiormente sul risultato finale. L'analisi di sensibilità è mirata a risolvere la problematica esposta nella Sezione 1.3.4, ed le sue modalità sono riportate nella Sezione 2.8.1.

Un altro importante tipo di analisi che può essere svolto in maniera complementare all'analisi di sensibilità concerne casi in cui le quantità in input nel modello siano cambiate a causa di cambiamenti effettuati nella tecnologia (e nelle evidenze a supporto), oppure casi in cui gli analisti vogliano considerare l'opportunità di raccogliere nuove evidenze, valutando i costi dell'operazione. In questi casi, gli analisti devono misurare la vastità dell'impatto che i cambiamenti, effettivi o ipotetici, presentano sul grado di soddisfacimento degli obiettivi strategici. Questo tipo di analisi, descritta nella Sezione 2.7.2, è nota come analisi dell'impatto dei cambiamenti, ed è relativa alle problematiche discusse nella Sezione 1.3.5.

2.8.1 Analisi di sensibilità

L'analisi di sensibilità è tipicamente condotta utilizzando o la regressione multivariata per passi (*multivariate stepwise regression*) oppure i coefficienti di correlazione per ranghi di Spearman (*Spearman's rank correlation coefficients*) [33].

Nell'analisi di regressione, i coefficienti computati per ogni quantità in input (cioè per ogni distribuzione di probabilità di un obiettivo foglia) misurano quanto l'output (cioè il grado di soddisfacimento di un obiettivo) è sensibile ad una variazione di quel determinato input. L'attendibilità generale di un'analisi di regressione è misurata con un indice da 0 ad 1. Se il suo valore è troppo basso (< 0.5), una simulazione simile sullo stesso modello potrebbe condurre ad un differente ordinamento delle sensibilità dell'output agli input.

Nell'analisi di correlazione, il coefficiente di Spearman viene calcolato per ogni coppia di distribuzioni in input-output. Tanto maggiore è il valore del coefficiente, quanto più significativo è l'input per determinare il valore dell'output.

Considerando l'esempio della Figura 21 della Sezione 2.7.2, è possibile ordinare gli obiettivi foglia sulla base del loro impatto sull'obiettivo di livello di più alto. I risultati dell'analisi sono riportati nella Figura 23.

Rank	Name	Regression	Correlation
#1	Send safe acceleration to train in time	0.821	0.832
#2	Safe acceleration recieved by train	0.408	0.392
#3	Maintain accurate speed and position estimates	0.369	0.38
#4	Maintain change of train acceleration from command	0.041	0.007
#5	Maintain computation of safe acceleration from estimates	0.041	0.068

Figura 23: Risultati dell'analisi di sensibilità per l'obiettivo radice della Figura 21

L'analisi mostra come l'obiettivo foglia "Inviare il comando di decelerazione del treno entro il tempo utile" abbia l'impatto maggiore sul soddisfacimento dell'obiettivo radice. Ciò era facilmente intuibile dal fatto che tutti gli obiettivi foglia contribuiscono allo stesso modo per l'obiettivo radice, e tra loro "Inviare il comando di decelerazione del treno entro il tempo utile" è quello con il più ampio intervallo di confidenza, che varia da 0.98 ad 1.

2.8.2 Analisi dell'impatto dei cambiamenti

Modus definisce anche linee guida per l'analisi automatizzata dell'impatto dei cambiamenti verso il soddisfacimento degli obiettivi. Si definisce il seguente

processo: innanzitutto si simula una distribuzione di probabilità per ogni input utilizzando le distribuzioni aggiornate per i nuovi input. In seguito, si confrontano le nuove e le vecchie distribuzioni attraverso statistiche riepilogative come tabelle, grafici e *box and whisker plot*.

Anche in questo caso, si presenta un esempio applicativo di questo tipo di analisi facendo riferimento alla Figura 21 della Sezione 2.7.2. Si supponga che la distribuzione relativa per l'obiettivo foglia "Inviare il comando di decelerazione del treno entro il tempo utile" è cambiata in seguito al miglioramento delle evidenze a suo supporto e alla conseguente nuova elicitazione dei dati. I nuovi valori per la distribuzione sono ora dati da $a = 0.99, m = 0.995, b = 1$. La Figura 24 riporta una tabella riepilogativa in cui sono riportati i cambiamenti di alcune grandezze di interesse relative al soddisfacimento dell'obiettivo radice, come media, minimo, massimo, moda, mediana, deviazione standard, varianza, curtosi, asimmetria, 5o e 95esimo percentile.

Input	Output: Maintain worst case Stopping Distance										
	Mean	Min	Max	Mode	Median	StdDev	Var	Kurtosis	Skewness	5%	95%
Send safe acceleration to train in time (baseline)	0.97783894	0.965949714	0.992388785	0.978203118	0.977844	0.004887742	2.389E-05	2.798542095	0.112820084	0.969429314	0.986124754
Send safe acceleration to train in time (after change)	0.982777028	0.973243117	0.991340816	0.987955511	0.982591987	0.003375989	1.13973E-05	2.741894742	0.000375613	0.976959229	0.988614798

Figura 24: Tabella riepilogativa dell'impatto di un cambiamento

Le stesse informazioni possono essere riportate graficamente utilizzando grafici di confronto, come riportato in Figura 25 per media, deviazione standard, 5o e 95esimo percentile.

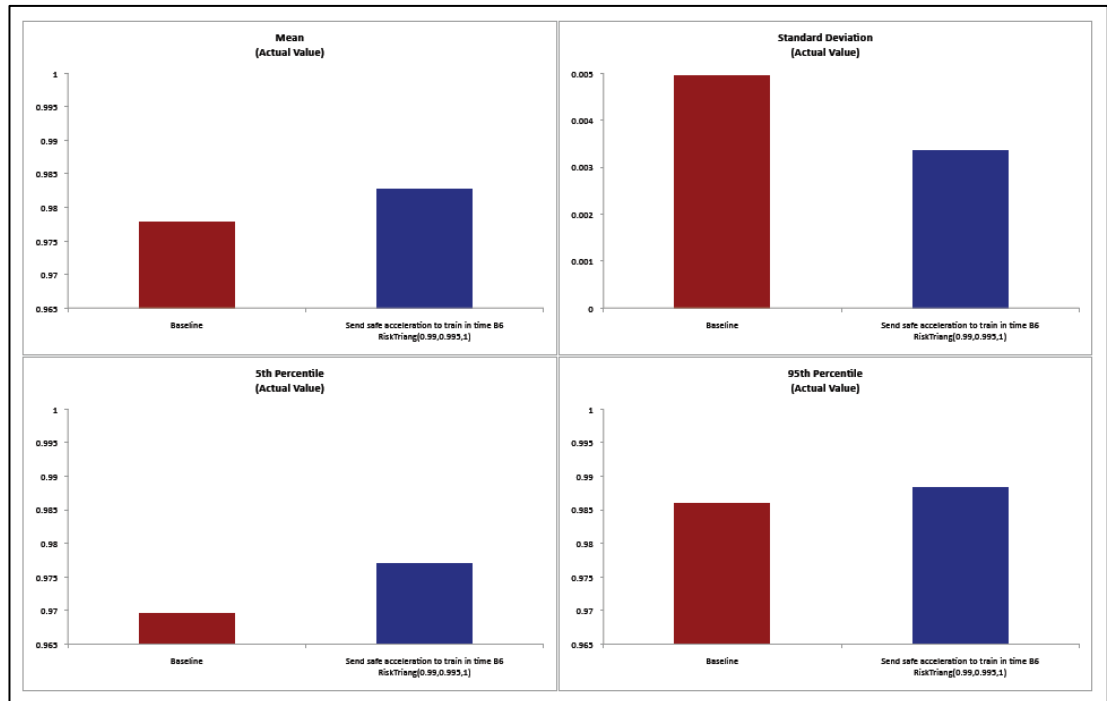


Figura 25: Grafici di confronto dell'impatto di un cambiamento

Un'altro strumento utile per analizzare l'impatto dei cambiamenti sono i box plot, come quello riportato in Figura 26.

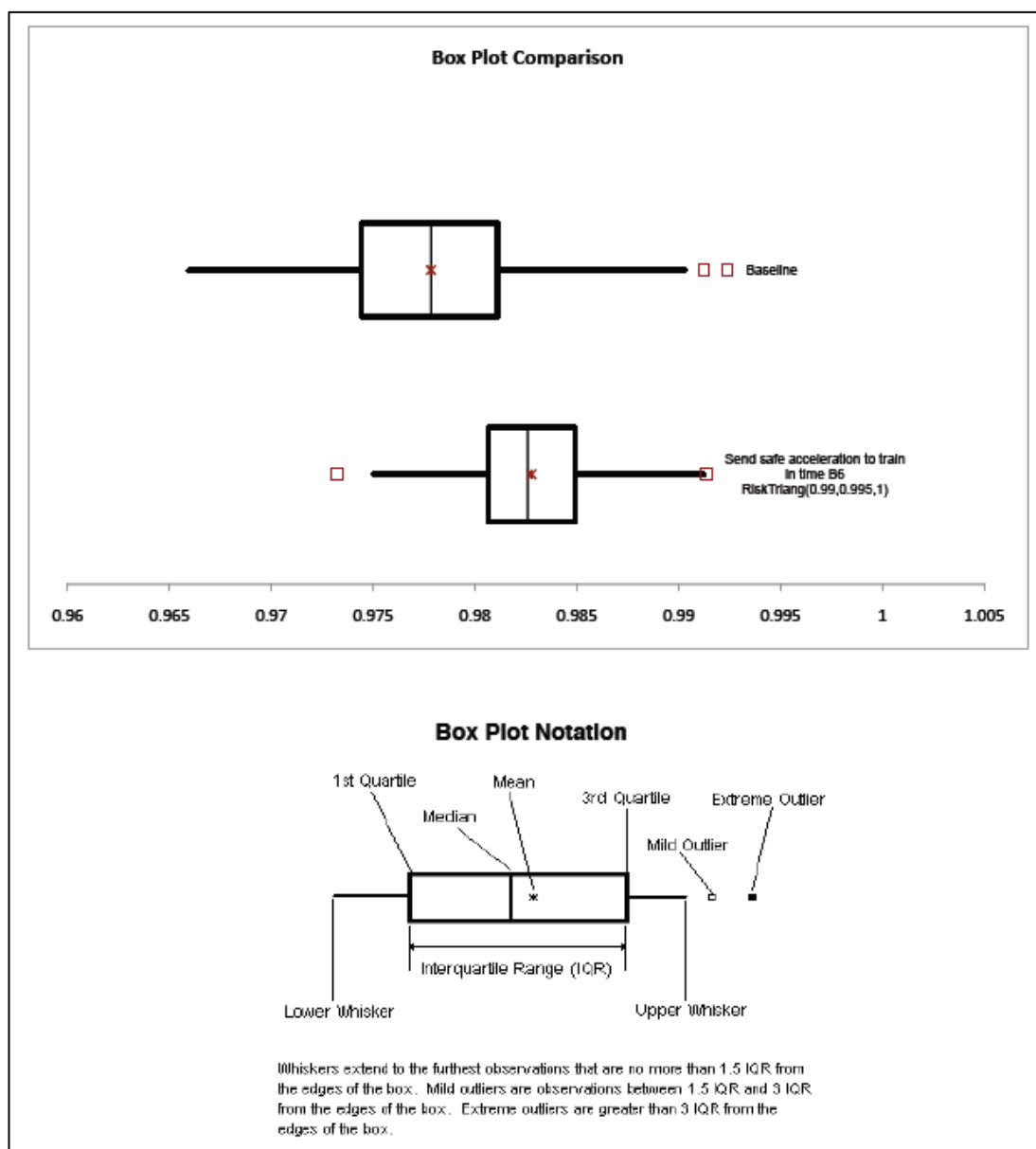


Figura 26: Box plot relativo all’impatto di un cambiamento

Analizzando i box plot, un analista può immediatamente notare che il cambiamento ha aumentato la probabilità di soddisfacimento dell’obiettivo radice. Riducendo inoltre il grado di incertezza relativo.

Nell’esempio presentato, si è considerata l’eventualità in cui solo un input fosse cambiato. Nel caso in cui si presentino simultaneamente più cambiamenti, è possibile scegliere tra due alternative per l’analisi dell’impatto dei cambiamenti: in primo luogo è possibile ripetere la simulazione per ogni input che è stato

modificato, analizzando separatamente i vari impatti. In alternativa è possibile considerare contestualmente tutti i cambiamenti, effettuando un'unica simulazione ed analizzando pertanto un solo insieme di dati.

Un'altro scenario di sicuro interesse per l'analisi degli impatti è il caso in cui gli analisti vogliano visualizzare come diverse valutazioni di una tecnologia evolvono quando uno o più dati in input evolvono. Si consideri a tal proposito la curva riportata nella Figura 22 della Sezione 2.7.2. Come detto in precedenza, se la probabilità di soddisfacimento desiderata dell'obiettivo "Mantenere la distanza di arresto relativa al caso peggiore" è del 98%, la curva riporta che è abbastanza improbabile (33%) che ciò avvenga.

Gli analisti potrebbero pertanto essere interessati a studiare come sarebbe possibile aumentare questa probabilità migliorando le evidenze, ad esempio utilizzando più esperti o conducendo più test. Ci si potrebbe infatti chiedere se migliorare le evidenze a supporto dell'obiettivo foglia "Inviare il comando di decelerazione del treno entro il tempo utile" possa essere di aiuto in tal senso. Nello specifico, si potrebbe voler sapere se aumentare il valore atteso m nella distribuzione triangolare associata a questo obiettivo impatta significativamente sul raggiungimento della probabilità desiderata. A tal fine è possibile computare le probabilità target attraverso delle simulazioni con diversi valori di m . I risultati delle simulazioni vengono in seguito riportati in diagrammi come quello della Figura 27.

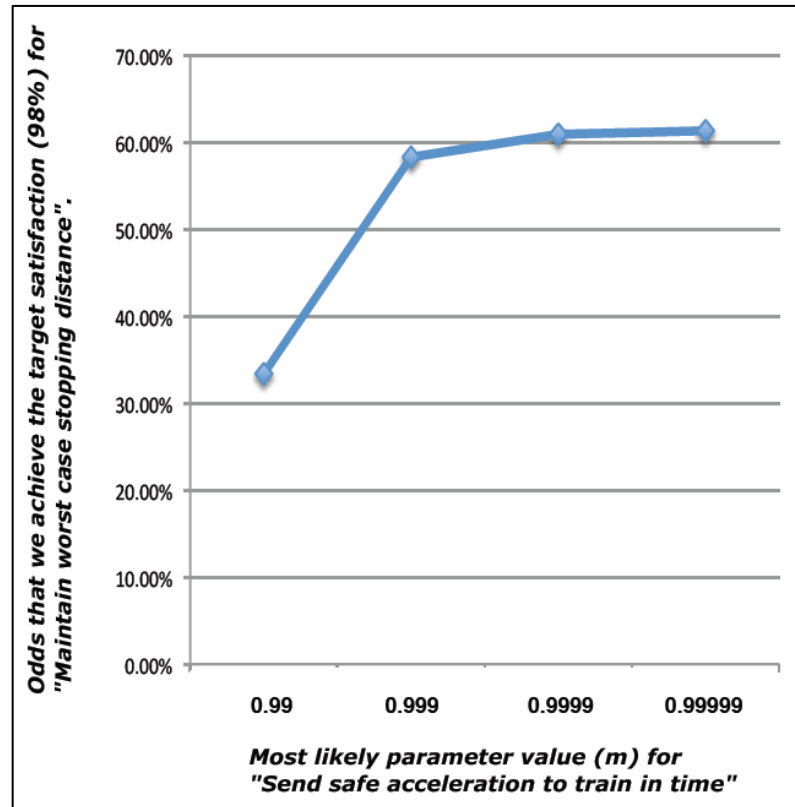


Figura 27: Analisi dell'impatto relativa all'evoluzione di un input

Il diagramma mostra come esista un ritorno significativo nel migliorare l'evidenza a supporto dell'obiettivo foglia in modo tale da aumentare il valore di m da 0.99 a 0.999. Tentare invece di raggiungere valori più alti di m sembrerebbe invece ingiustificato, in quanto il contributo per il raggiungimento della probabilità target oltre la soglia di 0.999 è molto esiguo.

3 Modus Tool: l'applicazione ideata e sviluppata

FAV_QUOTE_PLACEHOLDER

3.1 Obiettivi

La metodologia Modus per la qualificazione di nuove tecnologie è un processo che si articola in un intervallo temporale esteso e necessita la coordinazione di diverse figure professionali e la gestione di numerosi artefatti. Come ogni processo legato a progetti di grandi dimensioni, una gestione manuale risulta, dove semplicemente inapplicabile, quantomeno inefficace al conseguimento degli obiettivi del progetto stesso.

Modus Tool si configura pertanto come uno strumento a supporto della metodologia Modus, ponendo particolare accento su alcuni punti chiave del processo di qualificazione:

- Modellazione visuale efficace di goal e ostacoli a supporto della definizione del dominio del problema;
- Gestione delle attività svolte dalle diverse figure professionali che collaborano nel progetto, in termini di:
 - Separazione delle responsabilità operative legate ad ogni classe di utenti;
 - Organizzazione delle informazioni condivise;

- Facilitazione della comunicazione tra le parti.
- Gestione dei dati del progetto, in termini di:
 - Memorizzazione persistente dei dati in un formato condiviso e di pubblico dominio che facilita la rappresentazione dei dati su macchine diverse;
 - Controllo di consistenza sui dati stessi al fine di assicurarne l'integrità semantica;
 - Presentazione efficace di tutti i dati necessari all'attività di elicitazione delle opinioni di esperti, come dettagli di evidenze e modelli rappresentativi di goal e ostacoli.
- Interoperabilità con strumenti matematici necessari [6] per la valutazione quantitativa dei rischi legati alle tecnologie oggetto di studio.

3.2 Visione d'insieme

Si presenta di seguito una breve *overview* dell'applicazione sviluppata, le cui singole funzionalità saranno descritte nelle Sezioni 3.3, 3.4 e 3.5. Per una disamina di tutti i dettagli realizzativi si rimanda all'Appendice nella quale verranno presentati:

- La specifica dei requisiti utente (Appendice A);
- La specifica dei casi d'uso (Appendice B);
- La specifica dell'architettura (Appendice C);
- Il codice sorgente (Appendice D).

3.2.1 Glossario

Si riporta un glossario dei termini utilizzati nel corso del Capitolo 3. Laddove non fosse chiaramente interpretabile in altro modo dal contesto, vanno pertanto interpretati con il significato specificato nella Tabella 4. Per le definizioni dei concetti propri del goal modeling si rimanda al Capitolo 2.

Termine	Definizione
Decomposizione	<i>Vedi</i> relazione di decomposizione standard e relazione di decomposizione per riferimento
Distribuzione aggregata	<p>Distribuzione di probabilità calcolata componendo dati forniti da diversi esperti. È calcolata in modi diversi a seconda che essa rappresenti la composizione di valori puntuali o una distribuzione:</p> <ul style="list-style-type: none"> • Valori puntuali: valore della media aritmetica dei singoli valori puntuali • Distribuzioni di probabilità: distribuzione di probabilità della stessa classe avente come parametri i valori delle medie aritmetiche dei singoli parametri delle distribuzioni
Entità	Elemento di un goal diagram o decomposizione di un elemento in un goal diagram che rappresenta un evento su cui si desidera che l'esperto esprima un'opinione.
Entità "elicitabile"	Entità sul cui evento è possibile che un esperto esprima un'opinione.
Entità "elicitata"	Entità sul cui evento un esperto ha espresso un'opinione.
Elemento	Goal o ostacolo di un goal diagram
Elemento foglia	Elemento che non è decomposto in nessun altro elemento nel goal tree di riferimento
Elemento radice	Elemento che non è decomposizione di nessun altro elemento nel goal tree di riferimento
Elicitazione (struttura dati)	<p>Struttura dati che raccoglie le informazioni circa l'opinione di un esperto:</p> <ul style="list-style-type: none"> • Testo in linguaggio naturale riguardo il parere e valori di probabilità riguardo un evento • Sessione di elicitazione in cui l'opinione è stata espressa • Esperto che ha espresso l'opinione

	<ul style="list-style-type: none"> • Questionario di elicitazione a cui l'esperto ha risposto
Goal diagram	Diagramma di rappresentazione della decomposizione di un goal o un ostacolo secondo il formalismo presentato nel Capitolo 2
Goal model	<p>Modello che rappresenta tutti gli oggetti relativi ad un'attività di goal modeling. Comprende</p> <ul style="list-style-type: none"> • Un goal tree • Tutti i diagrammi in cui il goal tree è rappresentato • Tutte le entità che compongono il goal tree
Goal tree	Albero di decomposizione di un goal attraverso subgoal, ostacoli ed assunzioni. Può essere rappresentato in più goal diagram.
GUID	Global Unique Identifier. Identificatore univoco di un oggetto di EA all'interno del repository.
ID	Identificatore univoco di un oggetto di EA all'interno di Modus Tool.
Oggetto (di EA)	Struttura dati memorizzata nel repository di EA e contenente informazioni riguardo un qualunque artefatto di modellazione (entità, diagramma, progetto, folder, package...)
Progetto	Struttura dati contenente un insieme di goal model
Relazione di decomposizione standard	Relazione di decomposizione in cui gli elementi decomponenti sono rappresentati nello stesso diagramma della relazione stessa e dell'elemento decomposto.
Relazione di decomposizione per riferimento	Relazione di decomposizione in cui gli elementi decomponenti sono rappresentati in un diagramma diverso da quello della relazione stessa e dell'elemento decomposto e specificato nella decomposizione stessa.
Scope	Ambito di esistenza di dati all'interno di Modus Tool. Modifiche effettuate a dati di scope diversi sono indipendenti tra loro.

Sessione di elicitazione	Struttura dati contenente tutte le informazioni riguardo elicitazioni contestuali di entità contenute un goal diagram ed in tutti i suoi sottodiagrammi annidati: <ul style="list-style-type: none"> • Entità del goal model con il loro status di elicitazione; • Status di consistenza dei dati della sessione con le entità dei diagrammi della sessione.
Sessione inconsistente	Sessione di cui un diagramma è stato modificato successivamente all'ultimo salvataggio.
Sottodiagramma di un goal diagram	Diagramma il cui riferimento è parte di una decomposizione per riferimento di un elemento contenuto in un goal diagram.

Tabella 4: Glossario dei termini

3.2.2 Diagramma generale di attori e casi d'uso

Modus Tool è uno strumento pensato per supportare le attività di più figure professionali nell'ambito della qualificazione secondo Modus, tra cui:

- Esperti di dominio, che esprimono opinioni su eventi modellati in goal ed ostacoli;
- Amministratori o coordinatori dell'attività di elicitazione, che si occupano di modellare visualmente i goal, gestire i dati delle sessioni e mediare la comunicazione tra le parti coinvolte.

La Figura 28 riporta il diagramma degli attori di Modus Tool.

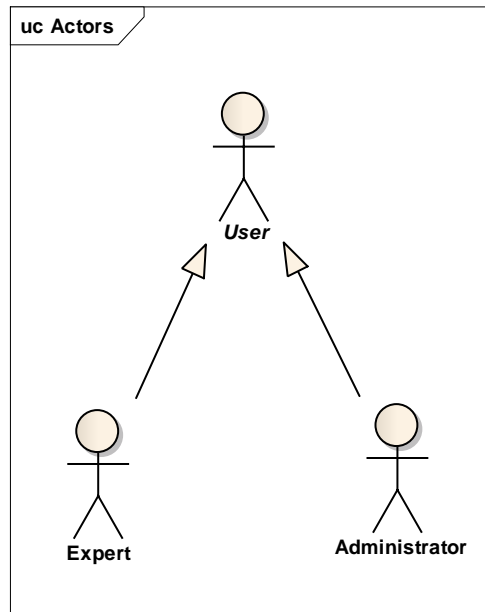


Figura 28: Diagramma degli attori di Modus Tool

Esperti ed amministratori hanno un insieme di responsabilità comuni (come ad esempio la visione dello stato della sessione di elicitazione), che sono identificate dall'attore astratto User generalizzazione di Expert ed Administrator.

In prima analisi, si possono individuare tre casi d'uso principali (Figura 29):

- Goal Modeling;
- Model Validation;
- Elicitation.

Una descrizione sommaria delle loro operazioni è riportata nella Sezione 3.2.3.

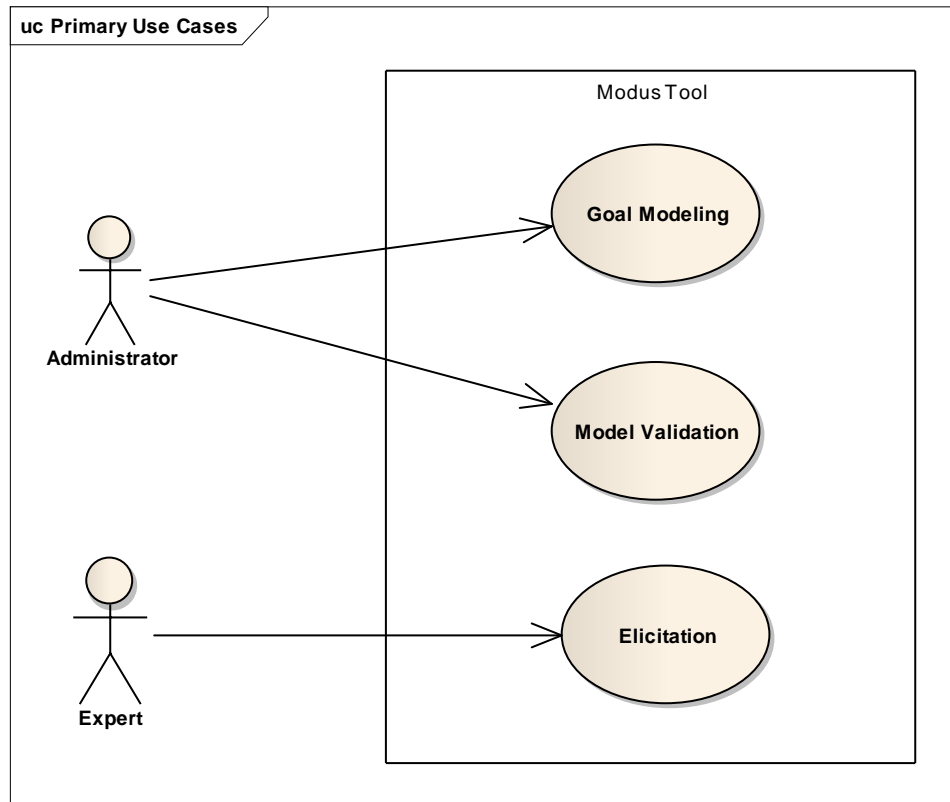


Figura 29: Diagramma dei casi d'uso principali di Modus Tool

Tuttavia, i flussi relativi alle tre funzionalità sono sufficientemente complessi da richiedere una disamina più esaustiva (Sezioni 3.3, 3.4 e 3.5). In fase di design i tre casi d'uso saranno dettagliati in numerosi altri casi sottocasi d'uso; tuttavia si manterrà questa definizione preliminare che ben rappresenta le macro-funzionalità offerte dal tool. Le specifiche dei casi d'uso, comunque presentate formalmente nella Sezione B, ometteranno per ragioni di sintesi espositiva il dettaglio di tutti i flussi relativi ad errori ed eccezioni, limitandosi alla descrizione dei soli flussi base e flussi alternativi principali.

3.2.3 Activity diagram generale

Come già presentato nel corso del Capitolo 2, un'attività di qualificazione di nuove tecnologie secondo Modus avviene in diversi *step* che conducono dalle specifiche del task alla valutazione finale delle performance.

La Figura 30 riassume l'attività in 5 marco-step principali:

- Goal Definition: definizione degli obiettivi relativi alla tecnologia e raccolta delle evidenze;
- Goal Modeling: modellazione dei goal in goal diagram attraverso decomposizioni in ostacoli ed altri goal;
- Model Validation: validazione strutturale del modello al fine di individuare eventuali errori di decomposizione;
- Elicitation: elicitazione dell'opinione degli esperti, sulla base di evidenze, su eventi elementari al fine di ottenere distribuzioni di probabilità di essi;
- Review: simulazione Monte Carlo al fine di propagare i valori delle probabilità degli eventi in tutti gli elementi dell'albero ed analisi dei risultati degli obiettivi principali della tecnologia.

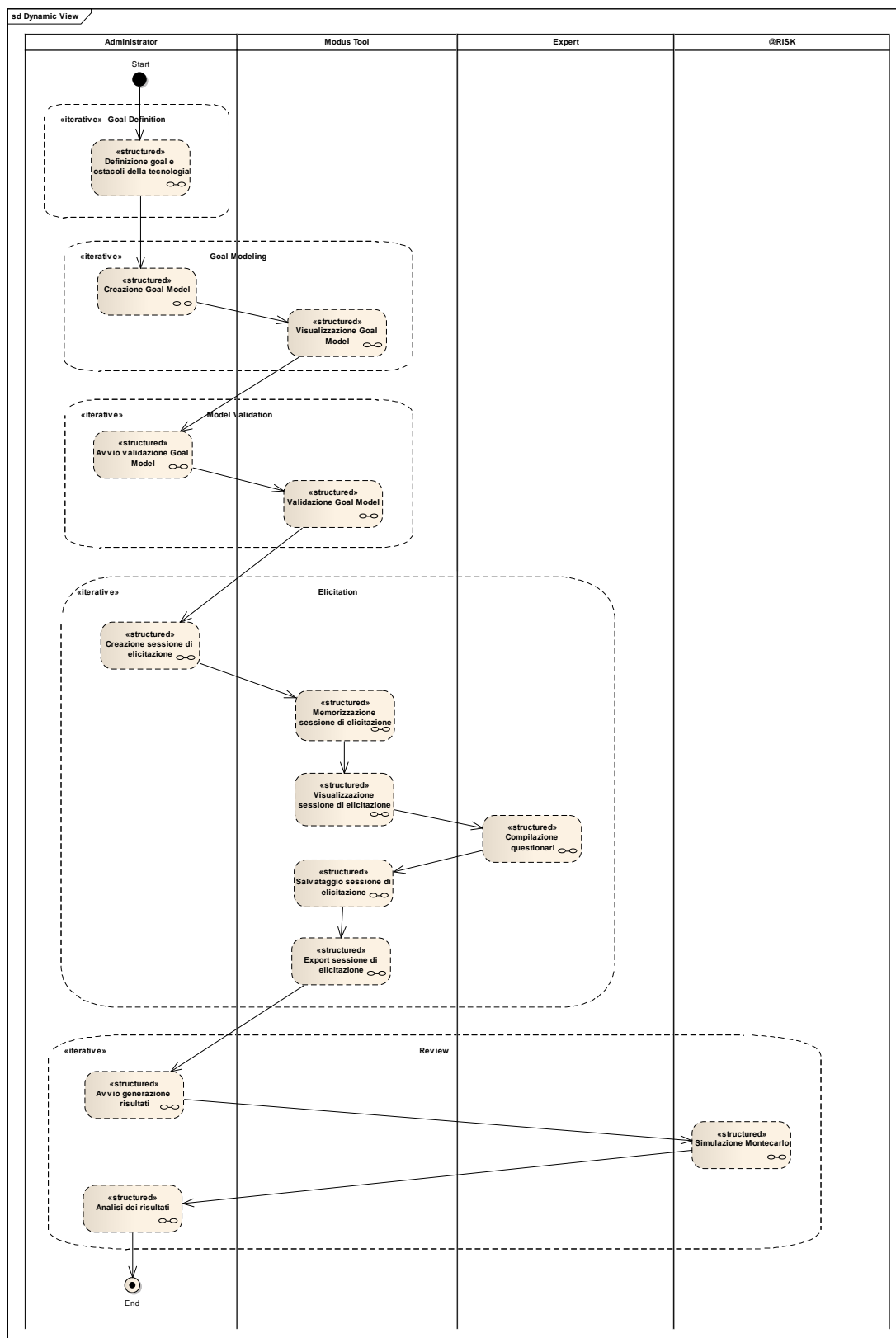


Figura 30: Activity diagram generale dell'attività di qualificazione di nuove tecnologie secondo la metodologia Modus

Modus Tool supporta i tre step centrali dell'attività di qualificazione secondo Modus, non offrendo supporto per l'attività di Goal Definition e limitando il proprio supporto in fase di Review alla sola interoperabilità con strumenti matematici per la valutazione quantitativa dei dati.

Il dettaglio delle attività supportate, analizzate sotto forma di casi d'uso, sarà oggetto delle Sezioni 3.3, 3.4 e 3.5.

3.2.4 Pattern architetturale BCE

L'architettura del sistema, realizzato come applicativo stand-alone, osserva il noto pattern architetturale Boundary-Control-Entity, in cui analogamente a MVC la rappresentazione dei dati presentati all'utente dalle Boundary è separata dalla gestione dei dati delle Entity in quanto il flusso delle operazioni è controllato dalle classi Control.

La Figura 31 riporta package e sottosistemi di Modus Tool, esplicitando le loro dipendenze.

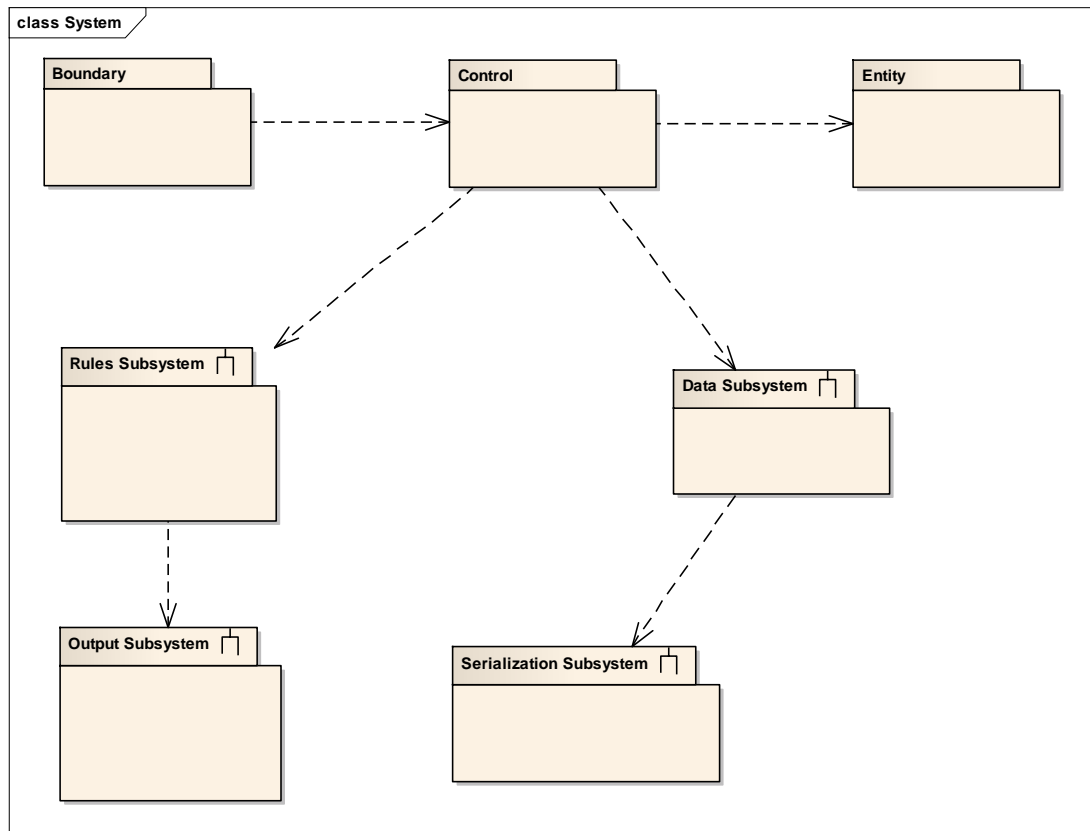


Figura 31: Overview architetturale di Modus Tool

Sono stati identificati 4 sottosistemi principali:

- Data, per la gestione della persistenza dei dati relativi a sessioni di elicitazione;
- Output, per la generazione di report testuali presentati nell'interfaccia utente nativa di Enterprise Architect;
- Rules, per il controllo delle regole di consistenza dei goal diagram;
- Serialization, per la serializzazione di dati strutturati.

Senza alcuna pretesa di esaustività, le Sezioni C.2.2 e C.3.1 descriveranno brevemente il modello architetturale di Modus Tool e dei suoi sottosistemi, rimandando ad altre sedi una disamina più approfondita delle scelte di design ed implementazione.

3.2.5 Enterprise Architect come piattaforma di riferimento

Sparx Enterprise Architect (EA) [5] è uno strumento per la gestione dei requisiti, l'analisi e lo sviluppo di sistemi software, per la modellazione del processo di business e per più generalizzati scopi di modellazione. Enterprise Architect consente una rappresentazione completa e al contempo intuitiva delle caratteristiche e dei requisiti di un sistema software, disponendo infatti, oltre che di una canonica rappresentazione testuale, anche di una modellazione grafica basata sui 13 formalismi definiti nella specifica di *UML 2.3* [34], tutti supportati dal tool. Enterprise Architect consente inoltre di definire in maniera molto intuitiva relazioni sia tra i requisiti che tra gli elementi di modelli diversi. Il tool non si limita alla sola gestione dei requisiti, ma supporta l'intero processo di sviluppo fino ad arrivare alla generazione automatica del codice in accordo agli standard definiti nella *Model Driven Architecture*, inclusa la possibilità di convertire diagrammi delle classi UML in formati standard *W3C XML Schema* secondo la specifica *XMI* [34]. *Forward e reverse engineering*, attuabili anche per e da molti *DBMS* commerciali, non sono le sole caratteristiche di Enterprise Architect: esso supporta la tracciabilità della documentazione di progetto, il *configuration management* e non ultima una serie di funzionalità di *groupware* atte a favorire la comunicazione e il lavoro parallelo dei diversi team di sviluppo impegnati su uno stesso progetto. In sintesi, Enterprise Architect è un tool che copre tutti gli aspetti del ciclo di sviluppo, fornendo piena tracciabilità dalla fase iniziale di analisi e progettazione fino al deployment e alla maintenance, e disponendo di solide possibilità di integrazione con altri IDE commerciali. Enterprise Architect presenta inoltre un framework di supporto per le *Model Driven Generation (MDG) Technologies*, con cui l'utente può importare risorse di una specifica tecnologia al fine di estendere le capacità di modeling del tool aggiungendo il supporto per notazioni e domini specifici.

La scelta di EA come *baseline* per l'implementazione di un add-in ha portato numerosi vantaggi tra i quali:

- Disponibilità di un prodotto general-purpose, flessibile, estensibile e maturo per il mercato dello sviluppo del software, essendo EA già ampiamente diffuso in ambito industriale;
- Utilizzo dell'intuitiva interfaccia grafica di cui l'ambiente è dotato per fornire all'utente un controllo semplice e immediato di tutte le funzionalità richieste;
- Implementazione di nuove funzionalità avanzate usufruendo della stabilità delle funzionalità di base e delle API offerte dall'ambiente, raccolte nella *Automation Interface (AI)* di EA;
- Fruizione dell'ampio set di entità e artefatti già presenti in EA e disponibili a diversi livelli di granularità;
- Facilità di estensione della feature di modeling attraverso la definizione di un'apposita MDG Technology;
- Abbassamento del *learning time* per gli utenti già familiari con il EA.

Enterprise Architect è stato inoltre utilizzato come tool di riferimento per la gestione di tutti gli artefatti relativi allo sviluppo dello stesso Modus Tool, dai documenti di specifica dei requisiti al design.

3.3 Caso d'uso Goal Modeling: dalla definizione alla sua implementazione

3.3.1 Definizione

La feature di Goal Modeling consente all'utente di realizzare un *goal model* contenente uno o più *goal diagram* relativi ad un *goal tree*, utilizzando l'interfaccia di modellazione fornita da Enterprise Architect.

Con questa funzionalità, Modus Tool ha l'obiettivo di fornire all'utente uno strumento completo ed intuitivo per la modellazione visuale di decomposizioni di goal ed ostacoli. I vantaggi di un approccio informatico alla modellazione di goal tree sono molteplici, e pongono le loro radici nell'incremento di produttività dato dalla semiautomatizzazione della gestione dei diagrammi. Soluzioni analoghe hanno ottenuto numerosi riscontri positivi, come nel caso dell'utilizzo dei tool di modellazione UML come lo stesso Enterprise Architect.

La struttura gerarchica in cui si organizza un goal model è analoga a quella di un modello standard in EA. Un modello radice (Root Model) contiene uno o più goal model; un goal model contiene uno o più goal diagram ed uno o più elementi presenti in un goal diagram. La Tabella 5 riporta la struttura di un goal model, rappresentata in modo analogo a quanto visualizzato nel *project browser* di Enterprise Architect.

Struttura	Descrizione
<ul style="list-style-type: none"> • Root Model 1 (root node) <ul style="list-style-type: none"> ○ Goal Model 1 (package) <ul style="list-style-type: none"> ▪ Goal Diagram 1 ▪ ... ▪ Goal Diagram n ▪ Element 1 ▪ ... ▪ Element n ○ ... ○ Goal Model n (package) 	<p>Root Model: modello principale relativo ad un progetto che contiene altri sottomodelli,</p> <p>Goal Model: modello contenente goal diagram, definito da Modus Tool</p> <p>Goal Diagram: diagrammi di decomposizione di goal e ostacoli, definito da Modus Tool</p> <p>Element: elemento contenuto in un diagramma, definito da Modus Tool</p>
<ul style="list-style-type: none"> • ... • Root Model n 	

Tabella 5: Struttura gerarchica di un goal model

L'utilizzo della funzionalità è del tutto analogo alla modellazione di diagrammi UML standard in Enterprise Architect: è sufficiente creare dal project browser un diagramma di tipo Goal Diagram all'interno di un Goal Model. In ogni goal diagram è possibile istanziare, tramite drag&drop dalla Toolbox di Modus Tool su un diagramma, elementi ed associazioni parte del formalismo di goal decomposition a partire dalle metaclassi e metassociazioni riportate nella Tabella 6.

Nome	Tipo
Goal	Metaclassa
Obstacle	Metaclassa
Assumption	Metaclassa
Evidence	Metaclassa
Complete AND	Metassociazione
Partial AND	Metassociazione
AND Branch Link	Metassociazione
AND Anchor Link	Metassociazione
Complete OR Link	Metassociazione
Partial OR Link	Metassociazione
Obstruction Link	Metassociazione
Assumption Link	Metassociazione
Evidence Link	Metassociazione
Artifact Link	Metassociazione
Reference Link	Metassociazione

Tabella 6: Metaclassi e metassociazioni della feature di Goal Modeling

Per ogni elemento, l'utente può inserire nel campo *Notes* (presente in EA per ogni elemento) delle note testuali contenenti informazioni sullo stesso.

3.3.1.1 Decomposizione di goal o ostacoli per riferimento

Un punto di forza della feature di goal modeling è la possibilità di articolare la decomposizione di un goal tree in più diagrammi. In contesti di reale applicazione,

gli alberi di decomposizione possono infatti svilupparsi su numerosi livelli, complicando la gestione di un diagramma monolitico. La complessità di un goal tree, intesa come numerosità degli elementi componenti, ha un impatto notevole sull'estensione dei diagrammi che li rappresentano, sicuramente maggiore rispetto a diagrammi le cui informazioni non dipendono dalla loro rappresentazione strutturale. In un *class diagram* UML, ad esempio, la disposizione degli elementi non è graficamente vincolata.

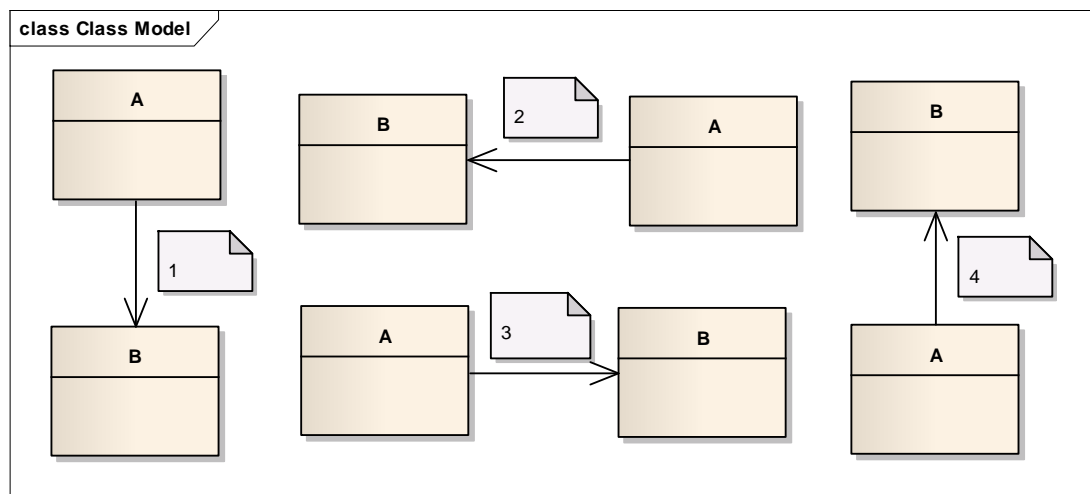


Figura 32: Disposizioni equivalenti degli elementi in un class diagram

Si consideri l'associazione diretta dalla classe A alla classe B (Nota 1) della Figura 32. Essa può essere equivalentemente rappresentata nel diagramma nelle configurazioni delle Note 2, 3 e 4.

D'altra parte un goal diagram, in quanto rappresentazione di un goal tree, vincola inerentemente la disposizione dei propri elementi ad una struttura ad albero.

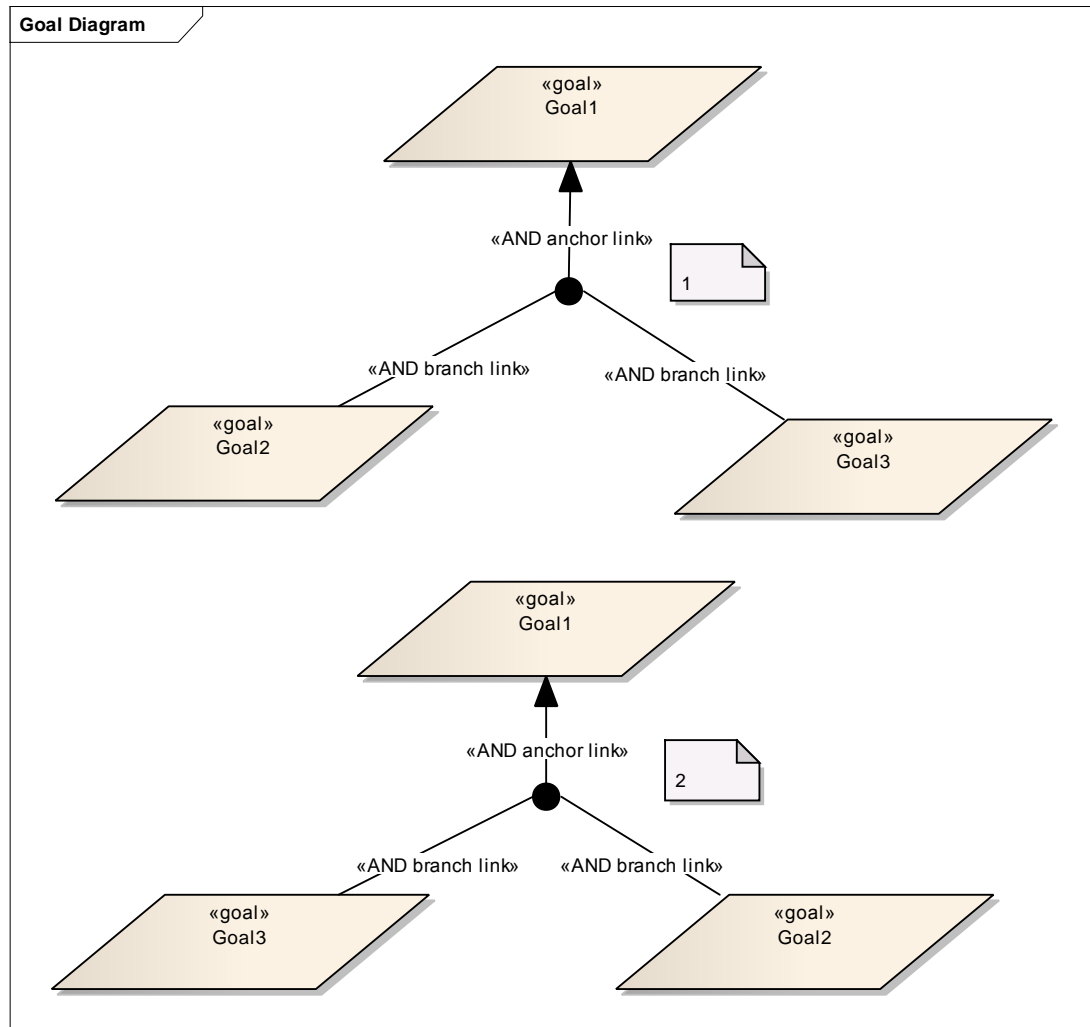


Figura 33: Disposizioni equivalenti degli elementi in un goal diagram

Si consideri ora la decomposizione attraverso un Complete AND di Goal1 in Goal2 e Goal3 (Nota 1) della Figura 33. Questa può essere equivalentemente rappresentata solo dalla decomposizione della Nota 2. È banale riscontrare infatti che una configurazione come quella della Figura 34 (Nota 3), non è corretta, nonostante essa sia in termini di elementi e connettori del tutto identica a quelle in Figura 33.

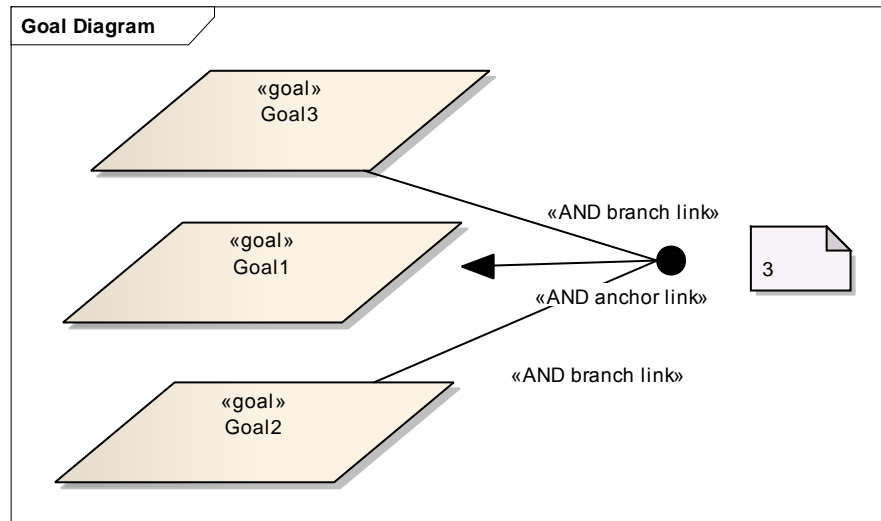


Figura 34: Disposizione errata degli elementi in un goal diagram

Ciò è causato dal fatto che in un goal diagram, essendo rappresentazione di un goal tree, si è per definizione obbligati a rappresentare gli elementi secondo una struttura ad albero che visualmente aiuta l'utente nella gestione della semantica dei dati.

Le condizioni che regolano la disposizione degli elementi in un diagramma possono essere formalizzate, effettuando un parallelismo con la meccanica classica, utilizzando il concetto di *gradi di libertà*. Si definisce numero di gradi di libertà di un diagramma il numero di variabili indipendenti necessarie per determinare univocamente la posizione di un elemento nel diagramma.

È immediato verificare che un class diagram ha 2 gradi di libertà, assumendo come non significativa la rotazione delle classi componenti attorno ad un piano perpendicolare al diagramma.

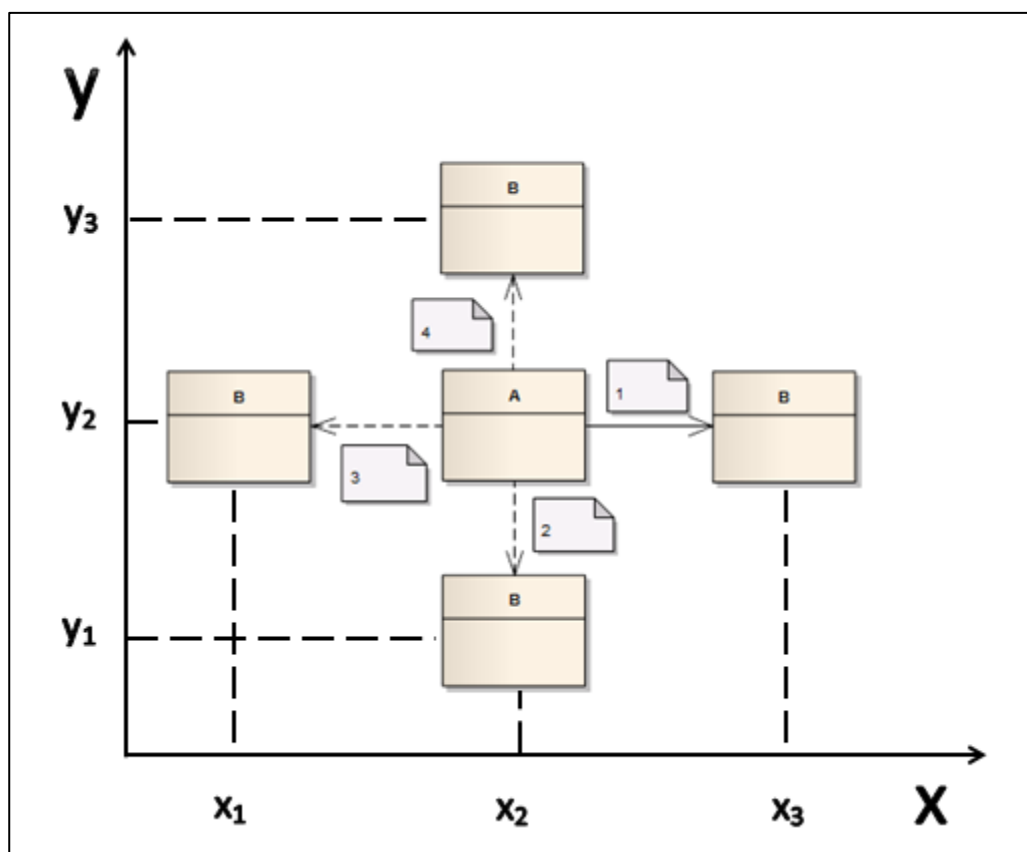


Figura 35: Gradi di libertà di un class diagram

Con la sola conoscenza del fatto che A presenta un'associazione con B sono richieste due variabili indipendenti (x ed y) per identificare la posizione relativa di B rispetto ad A nel diagramma: in Figura 35 sono esplicitate le coordinate della classe B sui due assi a seconda della sua posizione.

Un goal diagram presenta invece 1 grado di libertà (comunque trascurando la possibilità di rotazione degli elementi): la rappresentazione arborea dei dati forza a porre gli elementi decomponenti di un goal o ostacolo ad un livello di profondità inferiore a quello del padre. Ne risulta che, conoscendo la struttura del diagramma, la coordinata di ogni elemento è espressa da una sola variabile indipendente, cioè la posizione sull'asse orizzontale (che deriva dalla banale proprietà commutativa delle relazioni logiche di AND ed OR). La posizione sull'asse verticale è infatti deducibile dalla gerarchia di decomposizione dell'elemento.

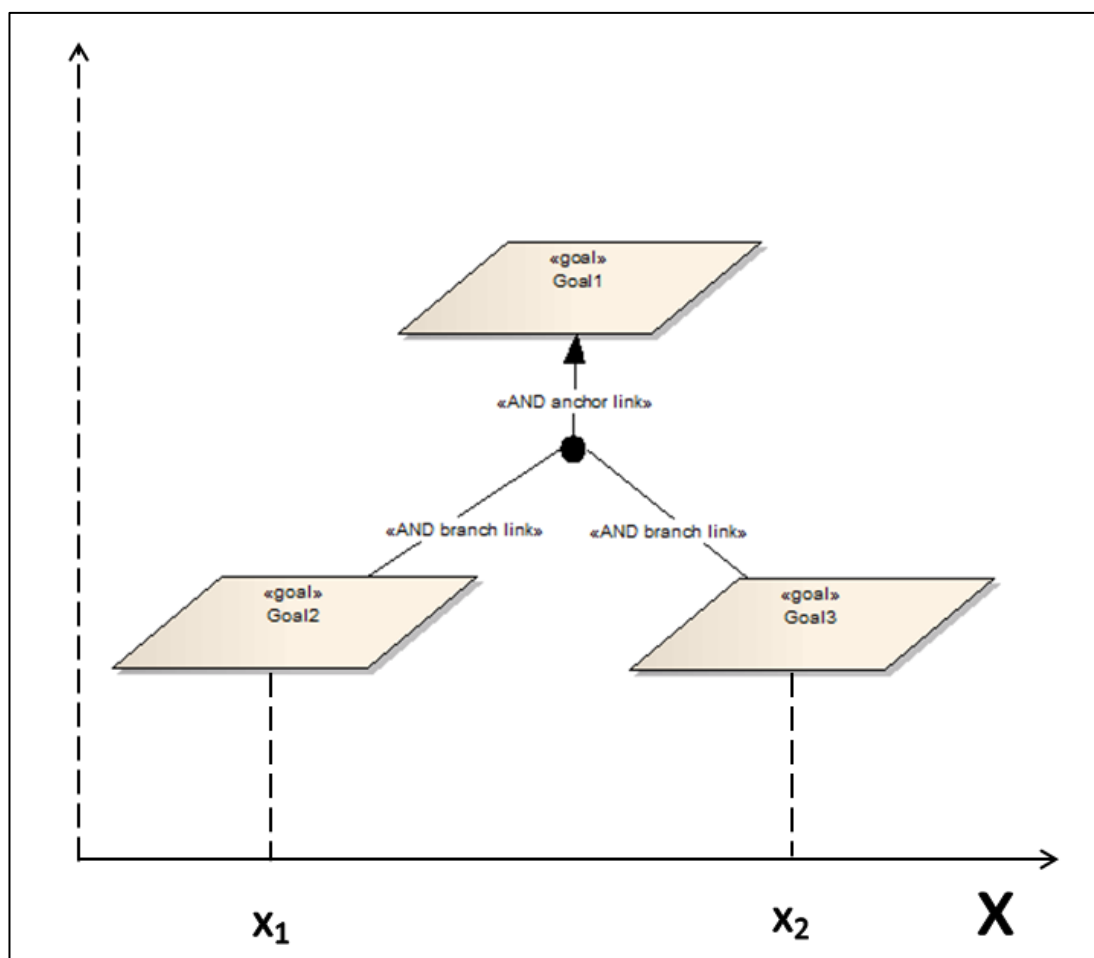


Figura 36: Gradi di libertà di un goal diagram

Nel goal diagram in Figura 36 è possibile notare che la conoscenza del fatto che Goal1 si decompone in Goal2 e Goal3 rende superflua la coordinata y della Figura 35, in quanto sicuramente Goal2 e Goal3 saranno posizionati sotto Goal1. Pertanto x è l'unica variabile indipendente richiesta per identificare la posizione relativa di Goal2 e Goal3 rispetto a Goal1.

Un minore numero di gradi di libertà comporta giocoforza una maggiore estensione del diagramma, i cui elementi sono vincolati a disporsi in lungo delle direzioni privilegiate. Per sopperire al rischio di una diminuzione dell'efficacia di un approccio informatico alla gestione dei goal diagram, ci si è ispirati alla soluzione già collaudata degli *interaction use* dei *sequence diagram*, che soffrono anch'essi dello stesso problema: basti pensare al concetto di *lifeline*, che vincola le frecce

rappresentanti messaggi scambiati da oggetti ad essere posti più in basso di messaggi che temporalmente li precedono. Così come un interaction use rappresenta un riferimento di un sequence diagram all'interno di un altro sequence diagram, un *hyperlink* rappresenta un riferimento di un goal diagram all'interno di un altro goal diagram. Ne segue che attraverso l'utilizzo degli hyperlink un goal tree possa essere rappresentato in più goal diagram.

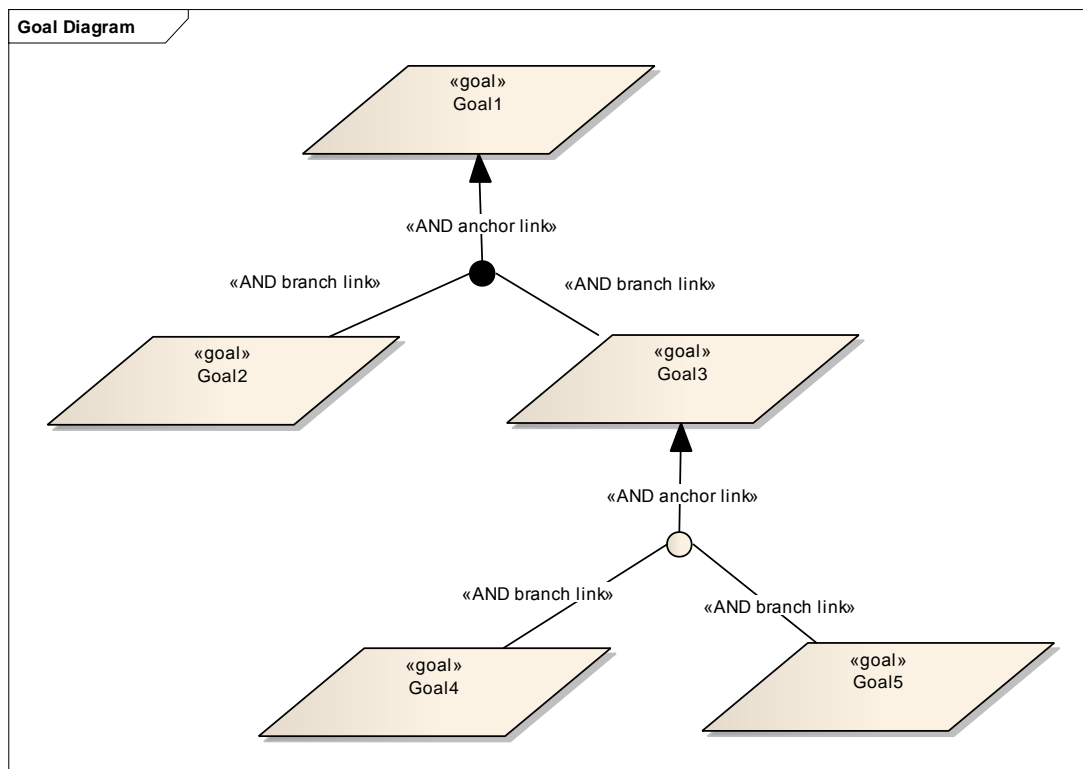


Figura 37: Goal tree decomposto in un singolo goal diagram

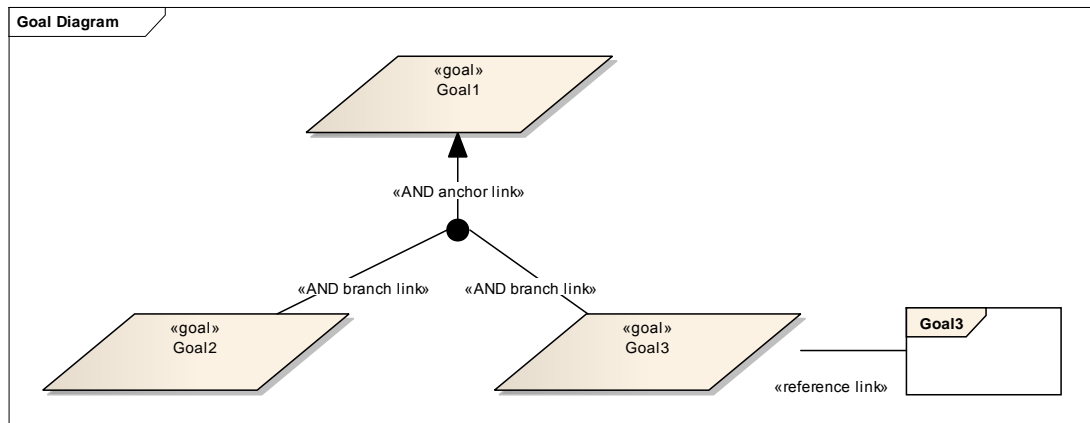


Figura 38: Goal tree decomposto in più goal diagram (diagramma principale)

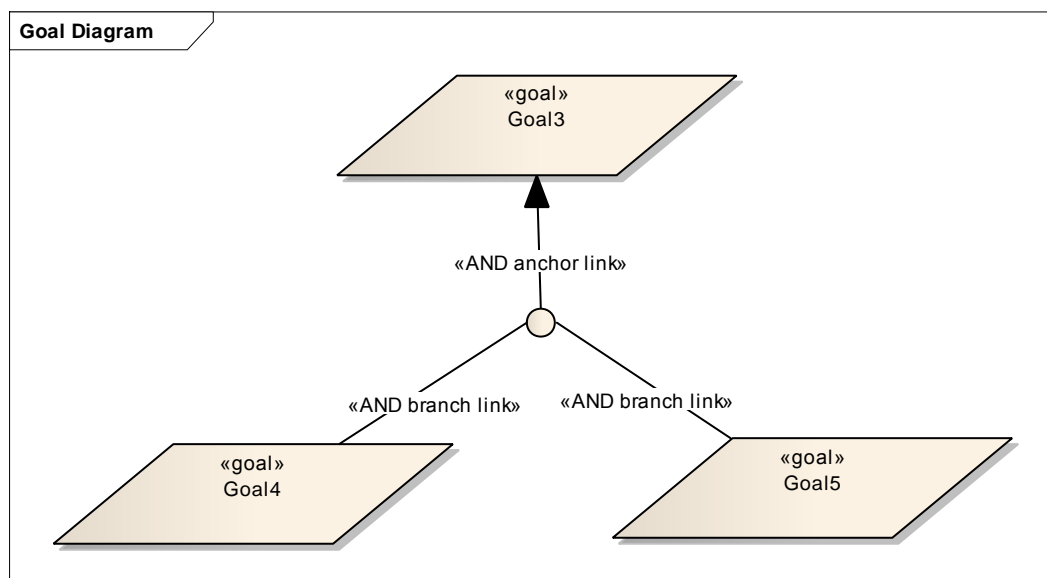


Figura 39: Goal tree decomposto in più goal diagram (diagramma secondario)

In Figura 37 è riportato un goal tree la cui decomposizione è interamente contenuta in un singolo goal diagram. In Figura 38 e Figura 39 si riporta lo stesso goal tree la cui decomposizione si articola in due goal diagram: in particolare la decomposizione di Goal3 in Goal4 e Goal5 viene referenziata nel diagramma principale attraverso l'utilizzo di un hyperlink.

L'utilizzo delle decomposizioni per riferimento aumenta di un'unità il numero di gradi di libertà di un goal diagram, per i quali è ora necessaria anche la variabile y che indica dove e se un elemento è decomposto (Figura 40).

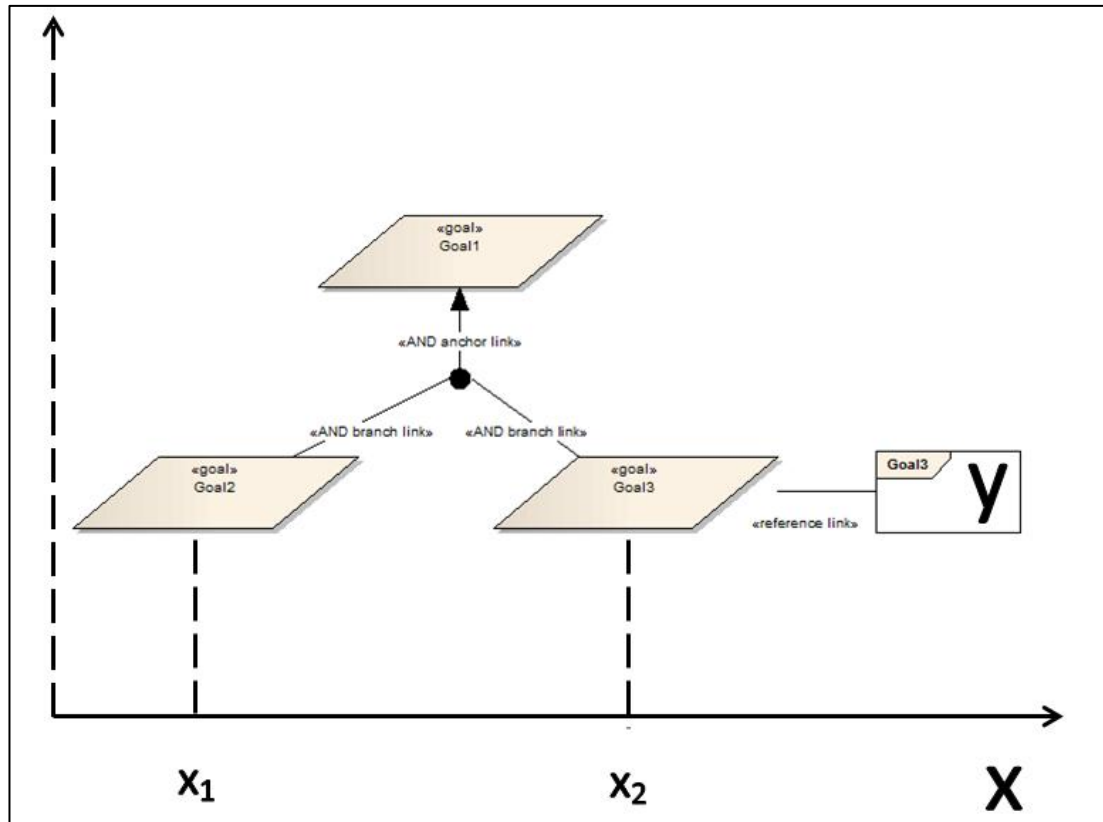


Figura 40: Gradi di libertà di un goal diagram con decomposizioni per riferimento

Ciò comporta una rappresentazione *multilayer* dei goal tree, che oltre a rappresentare in forma arborea decomposizioni di goal ed ostacoli, possono a loro volta essere rappresentati in alberi di decomposizione. Si considerino a tal proposito le decomposizioni della Figura 41.

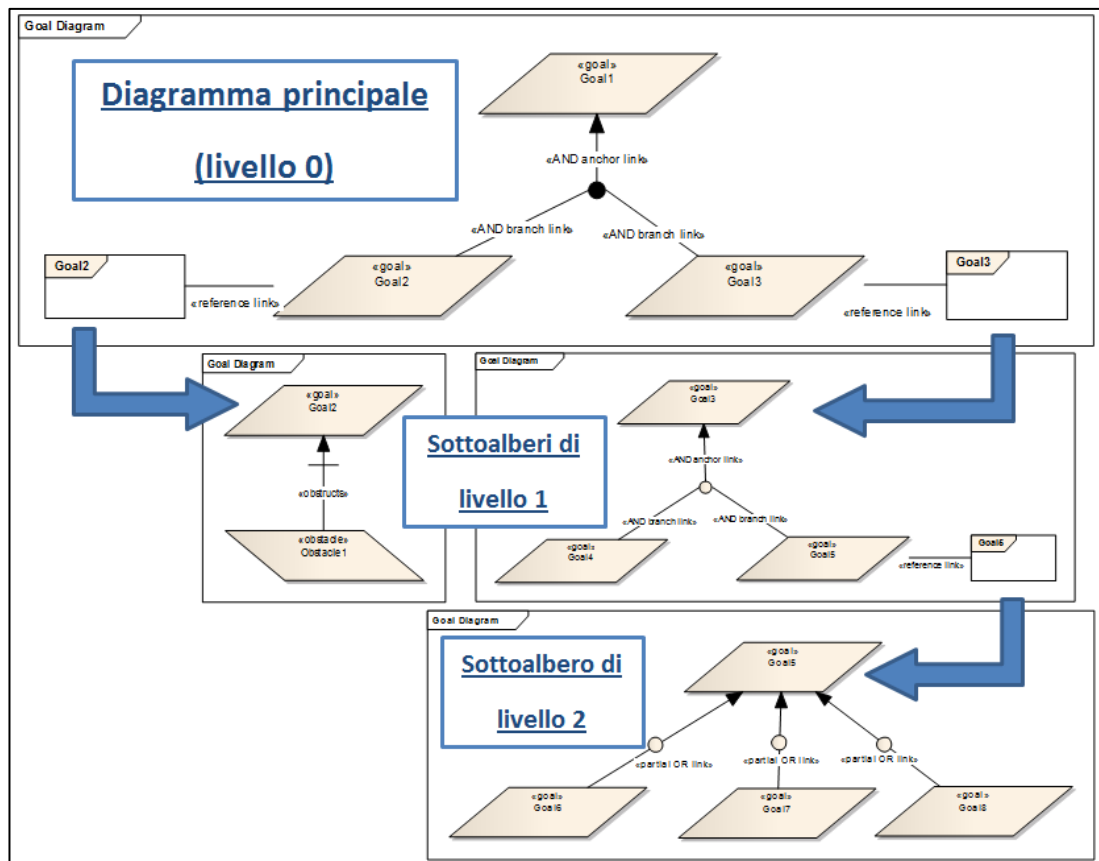


Figura 41: Struttura di decomposizione multilayer di un goal tree

Il diagramma principale presenta Goal1, Goal2 e Goal3, decomposti per riferimento in un altro diagramma. Il diagramma di Goal2 presenta la sua decomposizione in Obstacle1. Il diagramma di Goal3 presenta la sua decomposizione in Goal4 e Goal5, anche quest'ultimo decomposto per riferimento. Il diagramma di Goal5 presenta infine la sua decomposizione in Goal6, Goal7 e Goal8.

È pertanto possibile di individuare l'*albero dei diagrammi di decomposizione* in Figura 42.

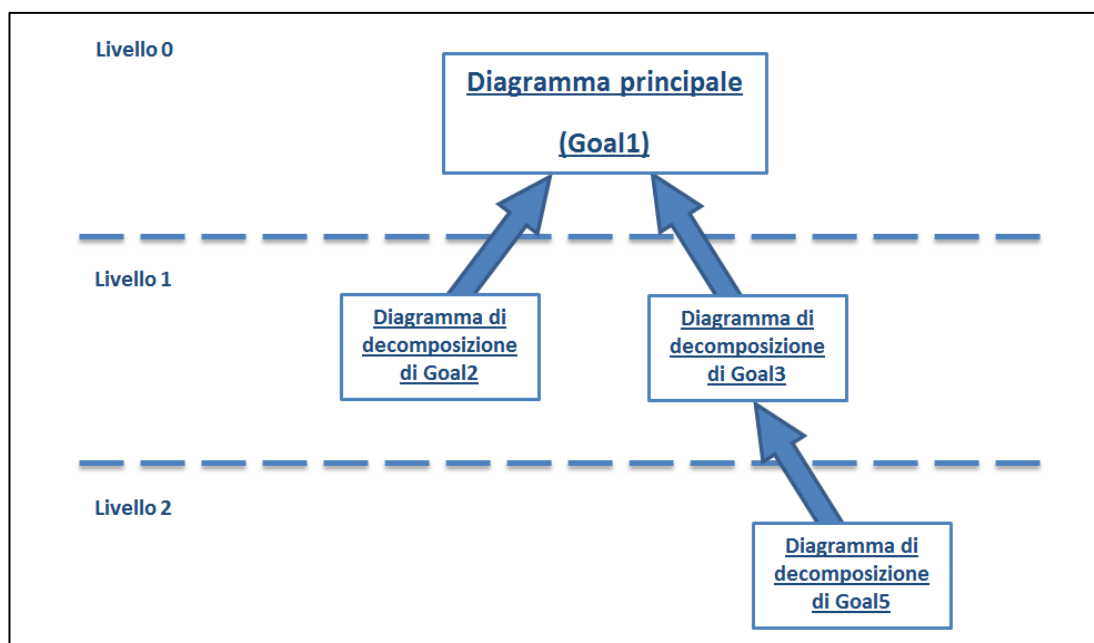


Figura 42: Albero dei diagrammi di decomposizione di Goal1

Tale definizione di una topologia di decomposizione non ha scopi meramente accademici: come verrà in seguito accenato a proposito delle attività che si basano sulla visita dei goal tree (Sezione 3.5.2.2.1), essa consente di definire soluzioni algoritmicamente eleganti ed efficienti.

3.3.2 Activity diagrams: specifica delle operazioni

L'utente accede alla funzionalità di Goal Modeling contestualmente alla creazione di un modello una volta avviata un'istanza di Enterprise Architect in cui sia stato specificato il caricamento dell'add-in. Un *workflow* ideale di un'attività di qualificazione di una nuova tecnologia inizia con le attività di creazione e visualizzazione di un goal model (Figura 43).

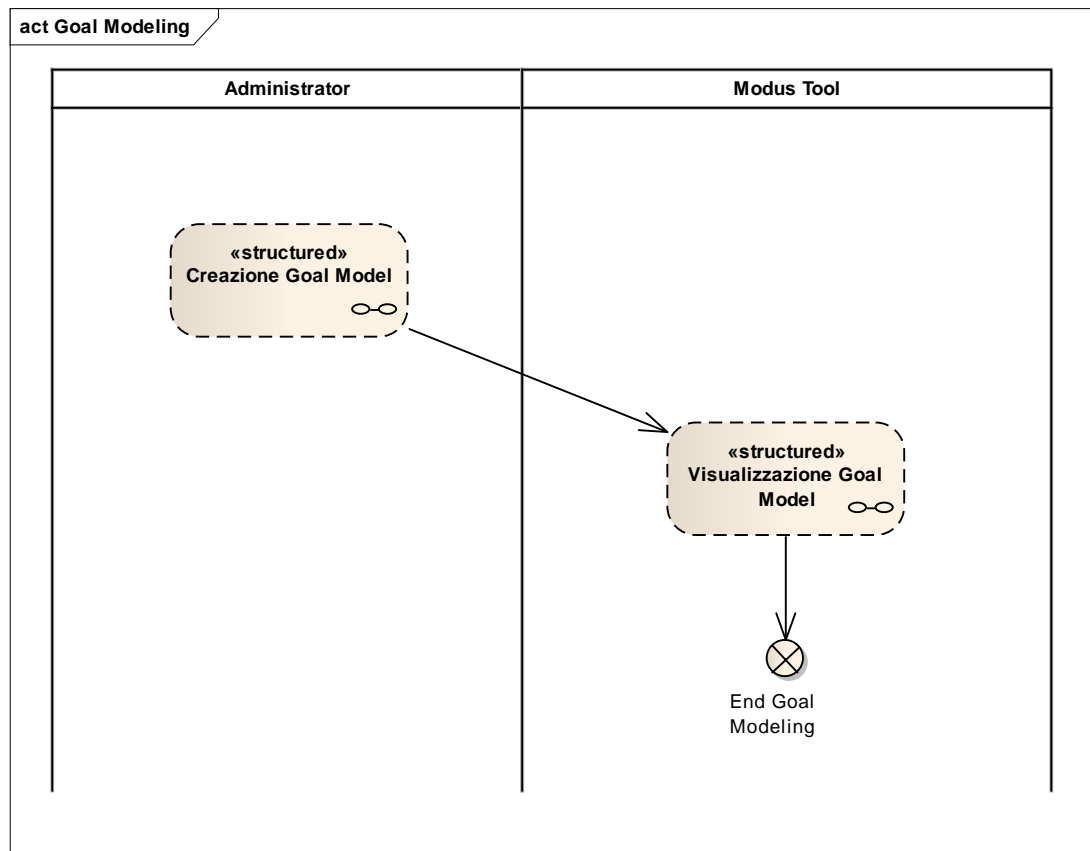


Figura 43: Activity diagram della feature di Goal Modeling

3.3.2.1 Creazione Goal Model

La Figura 44 riporta il dettaglio della *structured activity* relativa alla creazione di un goal model.

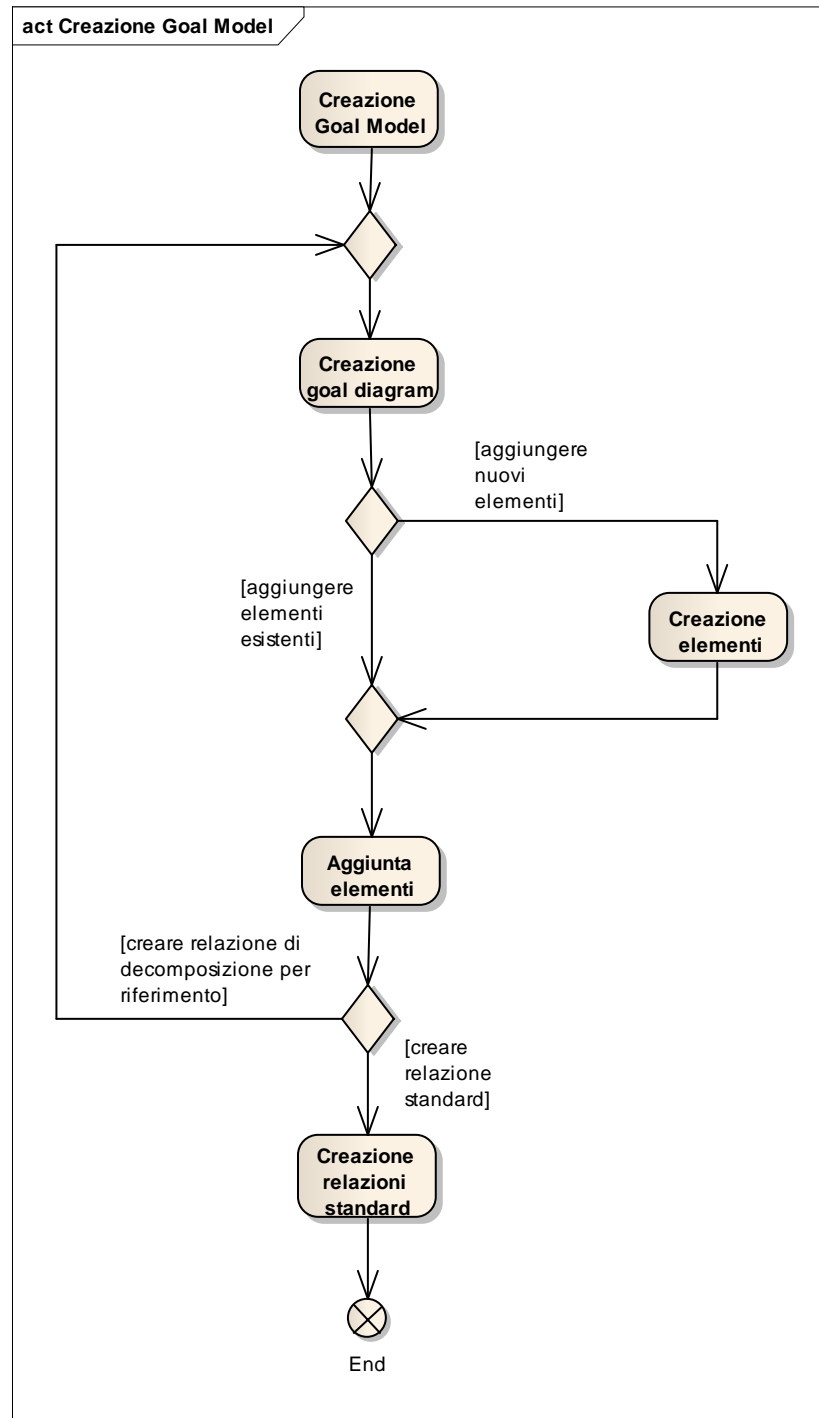


Figura 44: Structured activity "Creazione Goal Model"

Il processo di creazione è del tutto simile a quello di un qualunque modello UML. L'utente pertanto esegue nell'ordine le stesse operazioni che sono necessarie per un generico task di modeling:

- Crea un goal model
- Crea un goal diagram all'interno del goal model
- Aggiunge al goal diagram degli elementi, che possono essere
 - Creati ex novo con drag&drop dalla toolbox di Modus Tool
 - Già esistenti, ed aggiunti al diagramma con drag&drop dal project browser
- Crea tra gli elementi aggiunti delle relazioni, che possono essere
 - Relazioni di decomposizione di un elemento per riferimento, che richiedono la creazione di un nuovo goal diagram che specifica la decomposizione dell'elemento
 - Relazioni di decomposizione standard, in cui è sufficiente trascinare con drag&drop dalla toolbox di Modus Tool l'associazione corrispondente

Senza perdita di generalità, la modellazione dell'attività di goal modeling presenta alcune assunzioni. Per ragioni di chiarezza e sintesi espositiva, non è stata modellata nell'activity diagram la possibilità, comunque offerta da EA, di creare elementi non inseriti in alcun diagramma. Nonostante ciò sia lecito, si assume che in un flusso standard di creazione di un goal diagram un utente crei nell'ordine un modello, un diagramma, degli elementi e delle relazioni tra essi. Parimenti, il livello di granularità delle *actions* del diagramma impone la creazione prima di tutti gli elementi e solo dopo di tutte le relazioni tra esse, non specificando la possibilità, anche questa offerta dal tool, di creare elementi e relazioni in tempi intervallati.

3.3.2.2 Visualizzazione Goal Model

La Figura 45 riporta il dettaglio della structured activity relativa alla visualizzazione di un goal model.

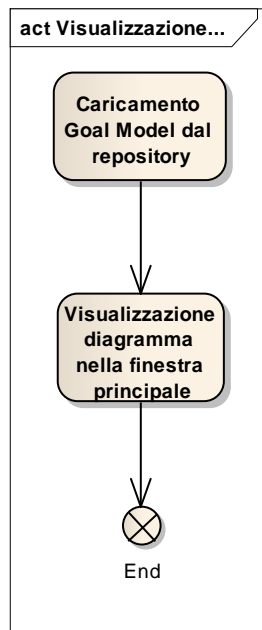


Figura 45: Structured activity "Visualizzazione Goal Model"

Le attività di visualizzazione sono completamente delegate all'interfaccia di Enterprise Architect, che carica il modello salvato dal repository e lo visualizza nella finestra principale.

3.3.3 Progettazione: MDG Technologies e profili UML

La feature di Goal Modeling è stata interamente realizzata con l'utilizzo delle MDG Technologies di cui alla Sezione 3.2.5. Il file MTS relativo alle definizioni delle notazioni di goal modeling, automaticamente caricato da EA all'avvio dell'add-in, è riportato in Tabella 7.

```

<MDG. Selections model="Modus_EA\Modus. eap">

    <Technology id="MODUS" name="ModusTechnology" version="1. 0"
    notes="Modus Technology for EA Modus Add-In"
    filename="Modus_Addin\Modus_Addin\Data\MDG\ModusAddinTechnology. xml"/>

    <Profiles directory="Modus_Addin\Modus_Addin\Data\MDG"
    files="ModusElementsProfile. xml"/>

    <TagVals tags="Modus: : Date, Modus: : Diagram, Modus: :
    DistributionParameters, Modus: : DistributionType, Modus: : Element, Modus: :
    Expert, Modus: : Session"/>

    <DiagramProfile directory="Modus_Addin\Modus_Addin\Data\MDG"
    files="ModusDiagramProfile. xml"/>
  
```

```

<UIToolboxes directory="Modus_Addin\Modus_Addin\Data\MDG"
files="ModusToolboxProfile.xml"/>

<ModelTemplates>
    <Model name="Goal Model" description="Goals, obstacles,
assumptions, evidences and their relationships" location="ModusGoalModelTemplate.
xml" default="yes" icon="34" filter="Goal"/>
</ModelTemplates>
</MDG. Selections>

```

Tabella 7: File MTS della Modus MDG Technology

I goal model sono stati definiti all'interno di EA specificando un file di metadati XML esportato secondo la specifica XMI (Sezione C.1.1). Le definizioni delle notazioni di goal modeling relative a diagrammi (Sezione C.1.2), entità (Sezione C.1.3) e toolbox di Modus Tool (Sezione C.1.5) sono contenute all'interno di profili UML che sono stati serializzati in un *MDG Technology Selection (MTS) file*, importato automaticamente all'avvio del tool. Per questi ultimi, sono unicamente riportati i diagrammi dei profili e non i rispettivi file XML contenenti le loro definizioni. Le forme delle entità sono state specificate con degli *Shape Scripts* associati agli stereotipi delle entità di un goal diagram ed interpretati dal motore di Enterprise Architect (Sezione C.1.4).

Si nota come il file MTS sia interamente e solamente composto dai riferimenti ai file XML contenenti le definizioni dei suddetti profili di cui ai paragrafi successivi.

3.3.4 Screenshots

Si presenta infine una verifica di funzionamento della feature di Goal Modeling, corredata da screenshot relativi agli step del flusso base della funzionalità che è stato presentato in forma di activity diagram nella Sezione 3.3.2.

3.3.4.1 Creazione di un Goal Model

Di norma, l'attività di Goal Modeling inizia con la creazione di un goal model e di un goal diagram (Figura 46), che secondo le opzioni di default è contestuale.

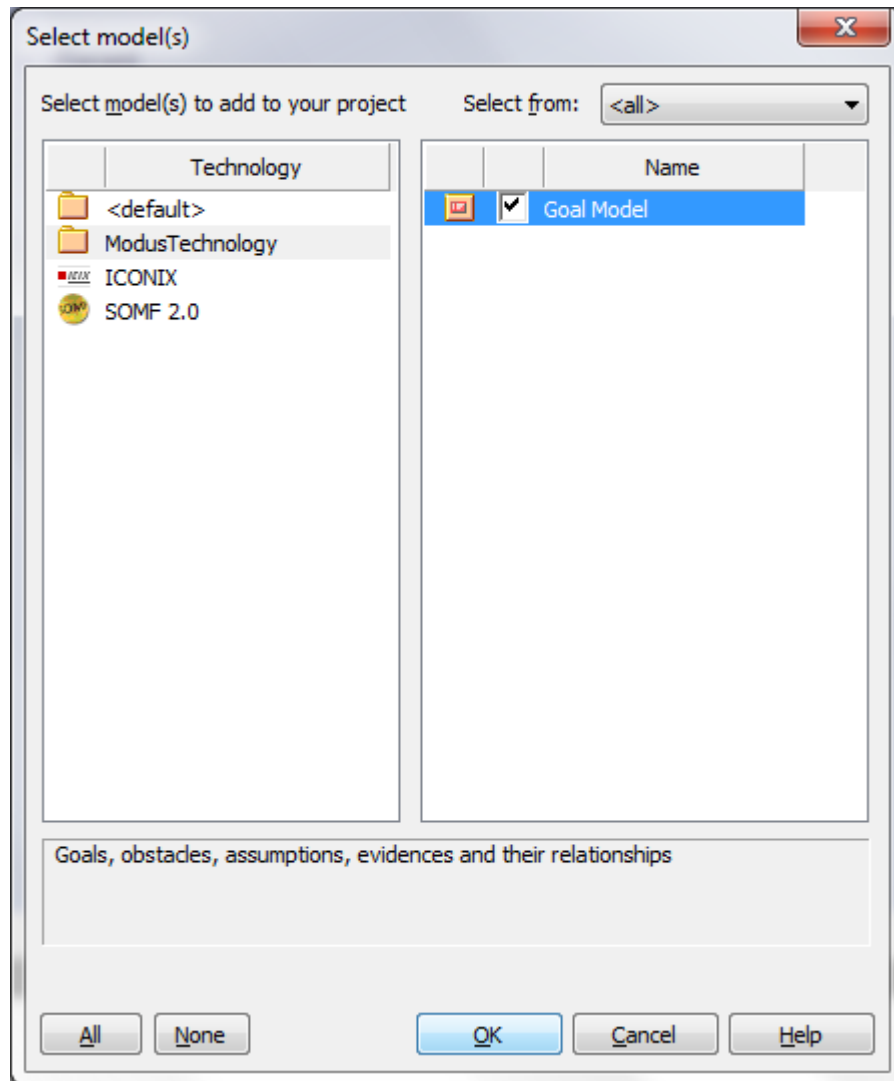


Figura 46: Creazione di un goal model

3.3.4.2 Creazione di un Goal Diagram

È comunque possibile creare un goal diagram indipendentemente dalla creazione di un goal model (Figura 47).

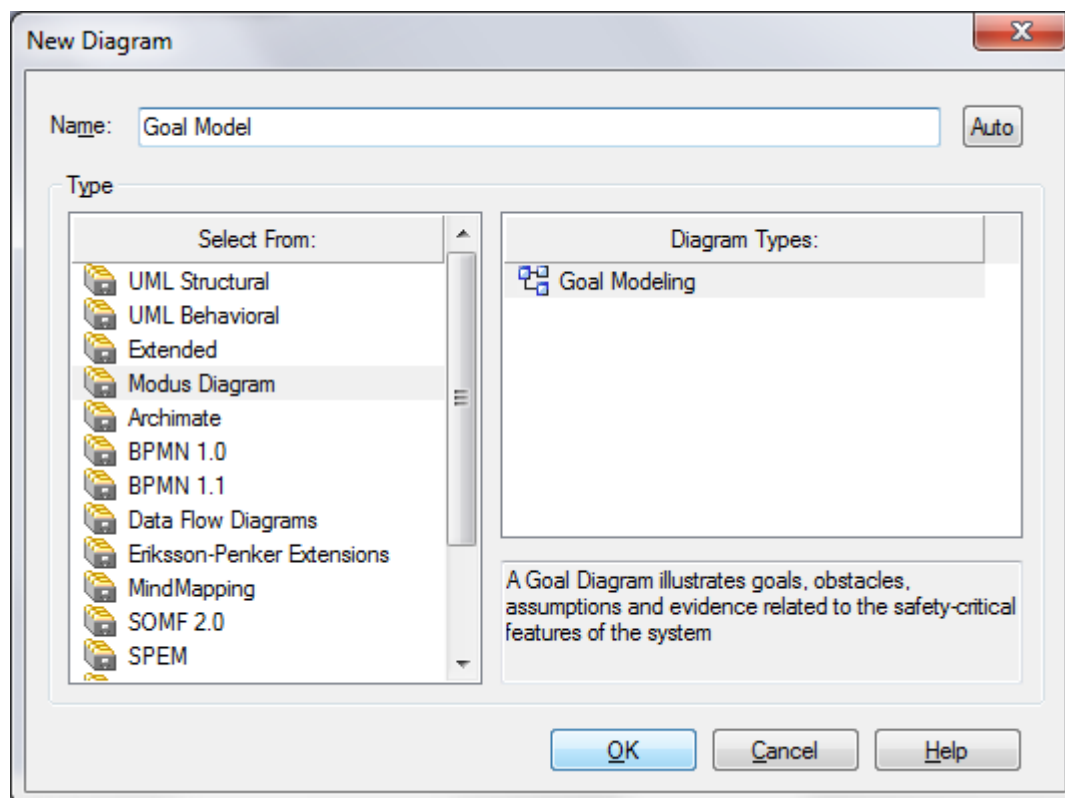


Figura 47: Creazione di un goal diagram

3.3.4.3 Creazione di un elemento

Dopo aver visualizzato un diagramma nella finestra principale, è possibile creare un elemento trascinando dalla toolbox il metaelemento corrispondente. Al rilascio del mouse, Enterprise Architect visualizza la finestra delle proprietà dell'elemento (Figura 48). Successivamente, l'elemento creato è visualizzato nel diagramma nella finestra principale, e nel project browser (Figura 49).

Goal : Goal1

General Details Requirements Constraints Links Scenarios Files Tagged Values

Name: Goal1

Stereotype: goal ... ☐ Abstract

Author: Stefano Status: Proposed

Scope: Public Complexity: Easy

Alias: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0 Advanced

Notes:

B I U A $\frac{1}{2}$ $\frac{1}{3}$ x^2 x_2

OK Annulla Applica ?

Figura 48: Creazione di un elemento

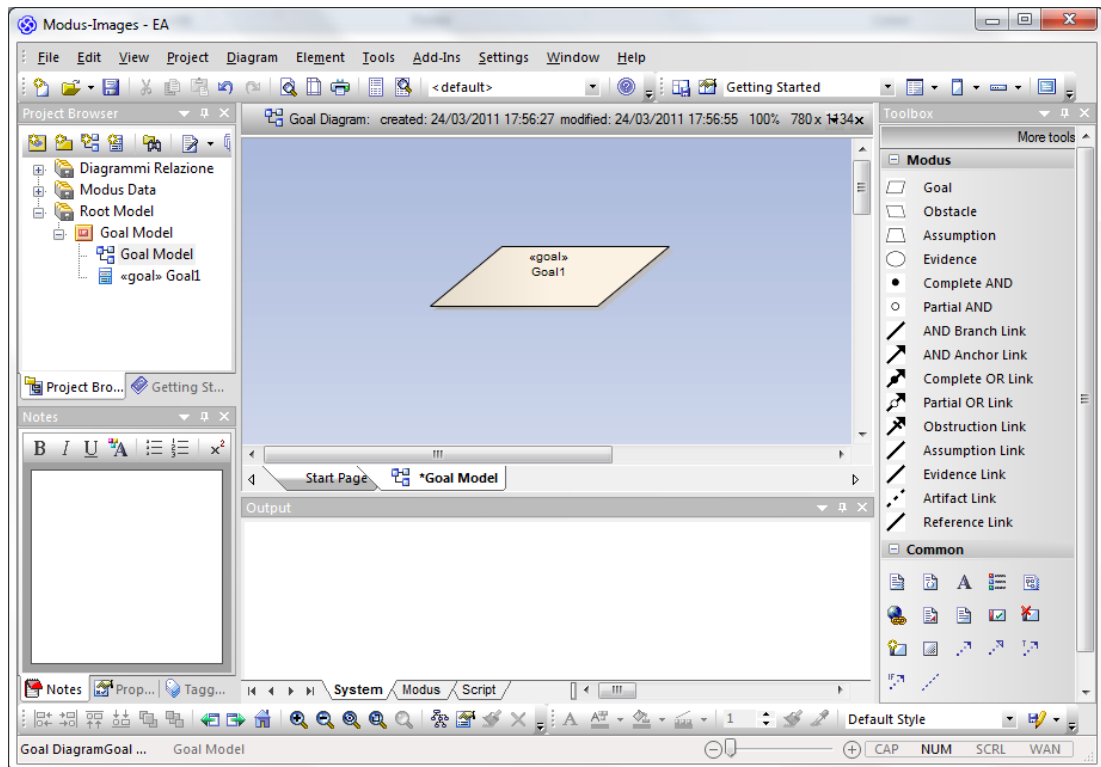


Figura 49: Visualizzazione dell'elemento creato nel project browser e nel diagramma

3.3.4.4 Creazione di una relazione standard

La creazione delle relazioni tra elementi è analoga alla creazione degli elementi stessi: è sufficiente trascinare lo stereotipo della relazione tra le due entità che si desidera collegare. La Figura 50 riporta la creazione di una relazione di ostruzione tra un goal ed un ostacolo.

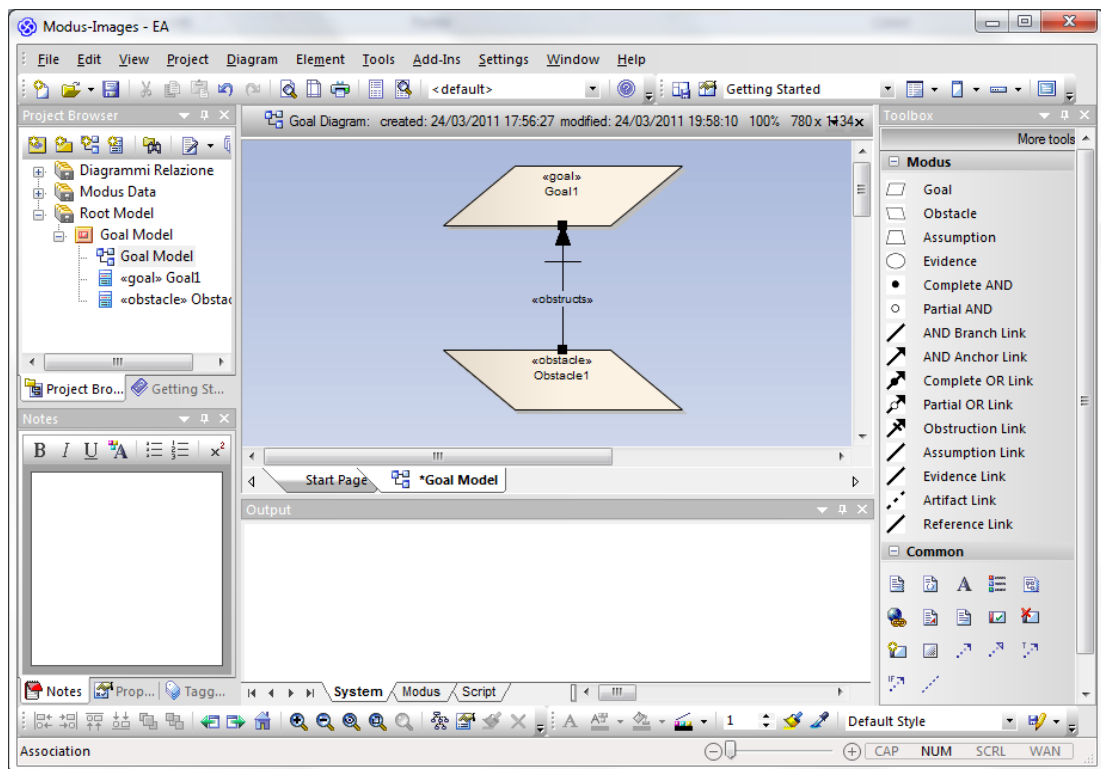


Figura 50: Creazione di una relazione standard

3.3.4.5 Creazione di una relazione di decomposizione per riferimento

La decomposizione per riferimento di un'entità prevede la creazione di un diagramma (Sezione 3.3.4.2) in cui saranno riportate le entità che decompongono l'elemento. Il diagramma contenuto viene pertanto trascinato dal project browser all'interno del diagramma contenitore come "Diagram Reference" (Figura 51).

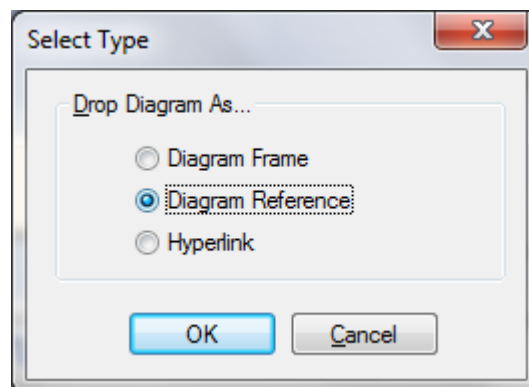


Figura 51: Creazione di una relazione di decomposizione per riferimento

Tra l'elemento decomposto ed il diagramma in cui la decomposizione è esplicitata viene quindi creata una relazione standard di tipo "Reference Link" (Figura 52).

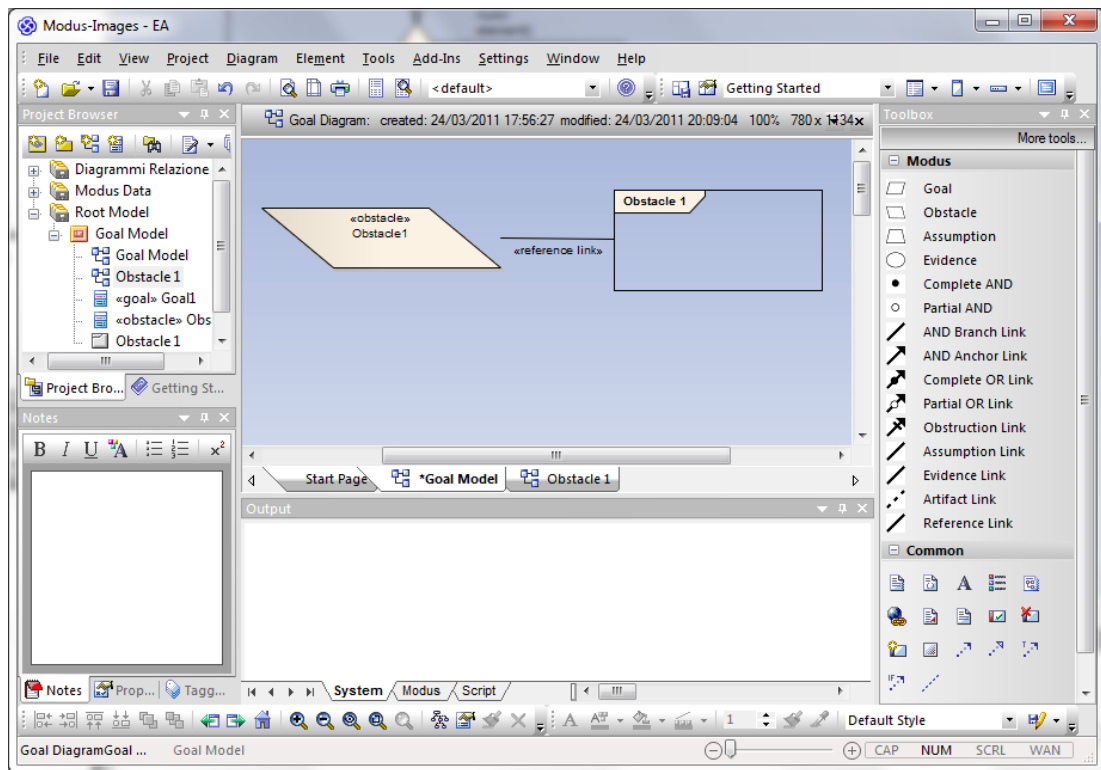


Figura 52: Visualizzazione di una relazione di decomposizione per riferimento

3.4 Caso d'uso Model Validation: dalla definizione alla sua implementazione

3.4.1 Definizione

La feature di Model Validation consente all'utente di validare un goal model in base a regole preimpostate, utilizzando l'interfaccia standard fornita da Enterprise Architect.

Con questa funzionalità, Modus Tool ha l'obiettivo di fornire all'utente la possibilità di verificare la correttezza strutturale di un goal tree, operazione resa non banale dalla complessità del dominio oggetto di studio. L'attività di individuazione degli errori commessi in fase di modeling è essenziale per assicurare la correttezza del modello stesso, e del risultato di ulteriori operazioni che si baseranno su di esso (Sezione 3.5). Il supporto fornito all'utente in fase di validazione, non presenta comunque funzionalità di correzione automatica, delegando all'utente la piena responsabilità di modifica del modello.

La validazione di un goal model avviene solo successivamente alla sua creazione, e sull'intero modello: non sono state sviluppate funzionalità di controllo dinamico (in tempo reale, contestualmente all'aggiunta di elementi ai diagrammi che compongono il modello), né di controllo mirato (su singoli elementi del modello, piuttosto che sul suo insieme). La funzionalità di validazione del modello è pertanto definibile come *reattiva*, piuttosto che *proattiva*.

Il controllo di consistenza effettuato sui goal model si basa su un insieme di regole preimpostato, che deriva dalla formalizzazione delle entità di goal modeling con un profilo UML (Sezione C.1.3). L'utente non ha la possibilità di modificare le singole regole che compongono l'insieme, ma può tuttavia scegliere quali regole validare: il compromesso attuato tra robustezza ed affidabilità della soluzione da una parte e flessibilità dall'altra sarà analizzato nel dettaglio nella Sezione 3.4.3.

Al termine di validazione, il tool presenta all'utente un report, indicando:

- Quante e quali regole sono state controllate su ciascun oggetto;
- Per ogni oggetto del modello, e per ogni regola che è possibile controllare sull'oggetto, l'esito del controllo della regola con informazioni aggiuntive ed eventuali suggerimenti per la correzione dell'errore;
- Statistiche di validazione: tempo impiegato, elementi controllati ed errori individuati.

3.4.2 Activity diagrams: specifica delle operazioni

L'utente accede alla funzionalità di Model Validation dal relativo menu di un'istanza di Enterprise Architect in cui sia stato specificato il caricamento dell'add-in. Il *workflow* di un'attività di qualificazione di una nuova tecnologia inizia con le attività di avvio ed effettiva validazione di un goal model (Figura 53).

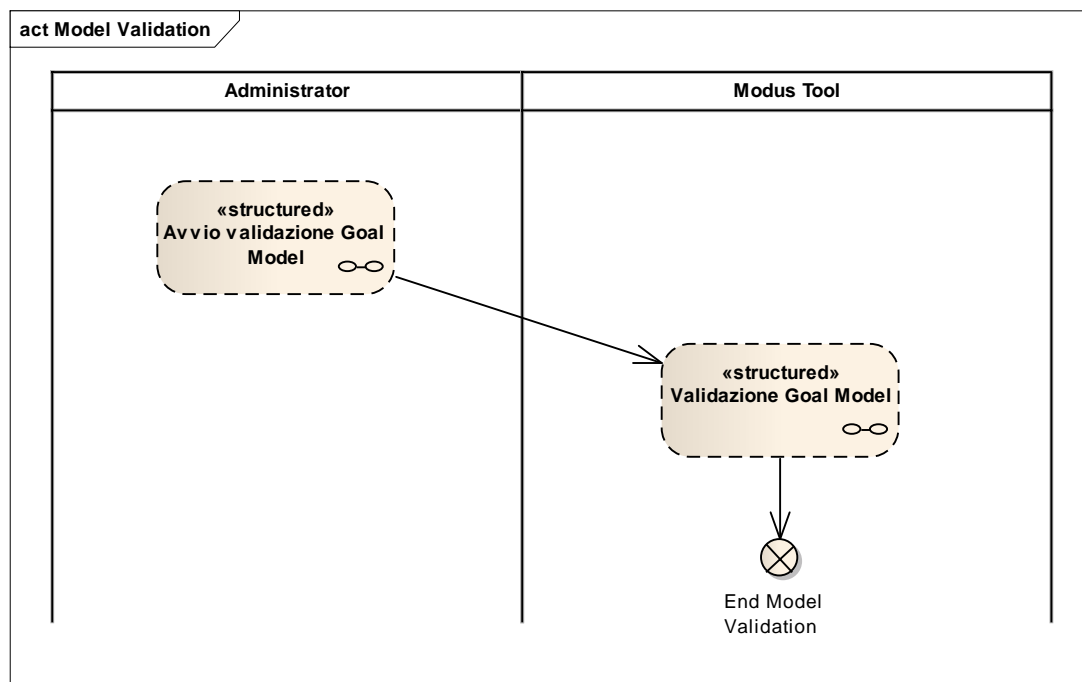


Figura 53: Activity diagram della feature di Model Validation

3.4.2.1 Avvio validazione Goal Model

La Figura 54 riporta il dettaglio della structured activity relativa all'avvio della validazione di un goal model.

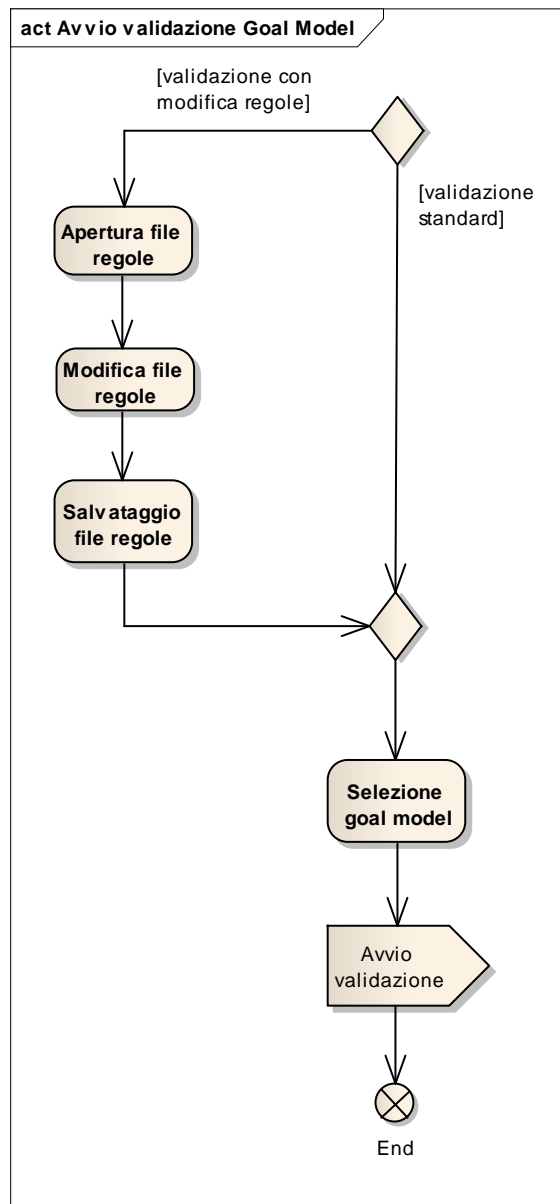


Figura 54: Structured activity "Avvia validazione Goal Model"

Opzionalmente, l'utente può scegliere di modificare il file contenente le regole specificando quali di esse saranno controllate durante la validazione. L'utente seleziona quindi un goal model ed avvia la validazione dal relativo menu di Enterprise Architect.

3.4.2.2 Validazione Goal Model

La Figura 55 riporta il dettaglio della structured activity relativa alla validazione di un goal model.

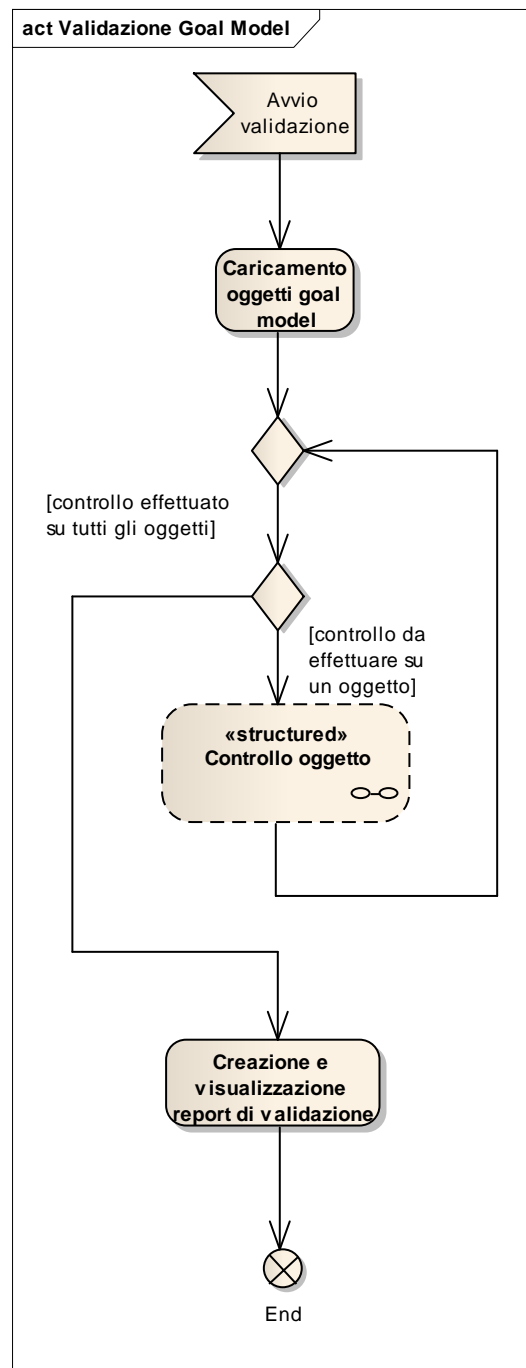


Figura 55: Structured activity "Validazione Goal Model"

In un primo momento, sono caricati tutti gli oggetti contenuti nel goal model. Successivamente, per ogni oggetto caricato viene effettuato il controllo delle regole (Sezione 3.4.2.2.1). Al termine della validazione, il sistema crea e visualizza un report della stessa contenente informazioni sugli errori individuati ed altre statistiche.

3.4.2.2.1 Controllo oggetto

La Figura 56 riporta il dettaglio della structured activity relativa al processo di controllo di un singolo oggetto.

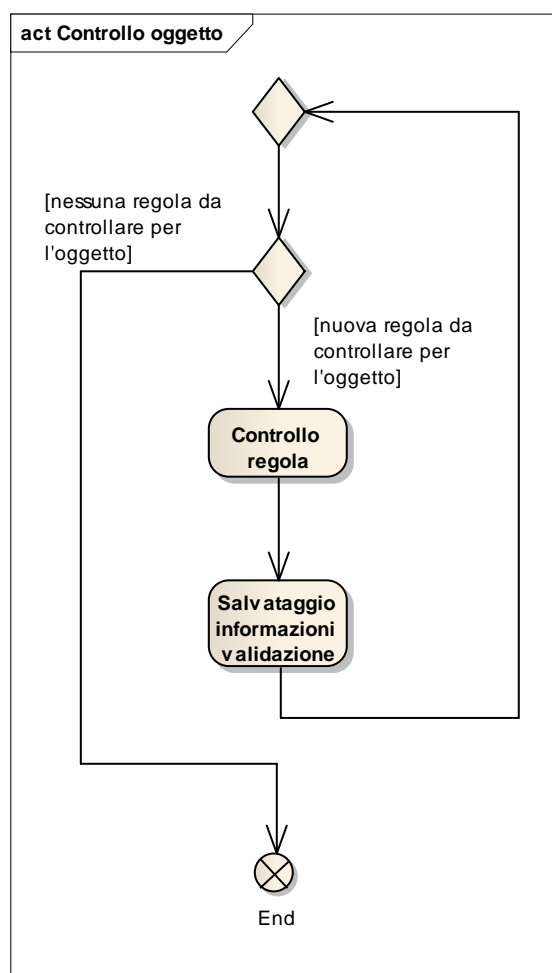


Figura 56: Structured activity "Controllo oggetto"

Non tutti gli oggetti dispongono di regole di validazione. Ciò potrebbe accadere per loro natura (Sezione A.2), oppure perchè l'utente ha deciso di non controllare le regole relative all'oggetto (Sezione 3.4.2.1). Durante l'attività di validazione gli oggetti per cui non è definita alcuna regola non vengono considerati. Per tutti gli altri oggetti il sistema effettua il controllo della regola e memorizza le informazioni riguardo lo stato di validazione, che saranno successivamente presentate all'utente.

3.4.3 Progettazione: regole di consistenza e framework di EA

La feature di Model Validation è stata realizzata utilizzando il framework di validazione di modelli offerto nativamente da Enterprise Architect. Un approccio di questo tipo, ponendo le proprie basi su una soluzione stabile e già operativa, ha contribuito alla realizzazione di una funzionalità che fosse quanto più possibile standard, manutenibile ed estendibile.

Tuttavia, la soluzione adottata risente dei limiti di flessibilità posti dalla stessa natura del framework proposto dall'EA di Enterprise Architect, come lo scarso supporto per la validazione real-time di un modello e la necessità di codificare l'implementazione delle regole a *compile-time*, limitando notevolmente le possibilità dell'utente di interagire con la semantica delle singole regole. In ogni caso, la soluzione adottata risponde appieno ai requisiti funzionali del sistema, secondo i quali non erano necessarie funzionalità avanzate di validazione.

Evitare l'utilizzo di approcci alternativi basati sull'esternalizzazione della *business logic* delle regole come l'utilizzo dei più flessibili *rule engines*, ha permesso allo studio in oggetto di vertere sulla definizione ad alto livello delle regole di consistenza, piuttosto che sul loro meccanismo di implementazione. Ciò è ulteriormente supportato dal fatto che l'infrastruttura di validazione di un goal model è strettamente correlata al formalismo che ne definisce la rappresentazione: la possibilità richiesta dall'utente di modificare le singole regole di consistenza

dovrebbe quindi essere giustificata dall'esistenza della possibilità di modificare il formalismo stesso (Sezione 3.3.3). Lo studio condotto non ha l'obiettivo di realizzare uno strumento di modeling universale, ma il suo scope resta confinato alla modellazione dei soli goal diagram, le cui specifiche si assumono raramente oggetto di modifiche nel tempo. Si delega pertanto ad eventuali estensioni future la possibilità di implementare meccanismi in grado di consentire all'utente di (ri)definire la semantica di regole di consistenza.

3.4.4 Screenshots

Si presenta infine una verifica di funzionamento della feature di Model Validation, corredata da screenshot relativi agli step del flusso base della funzionalità che è stato presentato in forma di activity diagram nella Sezione 3.4.2.

3.4.4.1 Modifica file delle regole

Prima di validare un modello, l'utente può opzionalmente scegliere di modificare il file contenente le definizioni delle regole, specificando di non eseguire il controllo di alcune di esse. La Figura 57 mostra la finestra di visualizzazione del file delle regole, in cui è possibile notare come i gruppi di regole siano stati descritti, oltre che dalle abstract rules, da dei commenti in linguaggio naturale.

```

Rules.csv - Blocco note
File Modifica Formato Visualizza ?
Index; Description; Error Message; Suggestion; Subrules

// syntactic (connectors) rules
1;The goal model must be syntactically correct;There exists at least a conn
2;Each AND branch link must connect either a goal or an obstacle to either
3;Each AND anchor link must connect a complete AND node or partial AND node
4;Each complete OR link must connect either a goal to another goal, or an o
5;Each partial OR link must connect either a goal to another goal, or an ob
6;Each obstruction link must connect either a goal to an obstacle, or an ob
7;Each assumption link must connect an assumption to a complete AND node;Th
8;Each evidence link must either connect a goal or obstacle to an evidence;
9;Each artifact link must connect an evidence to an artifact;There exists a
10;Each reference link must connect a goal or obstacle to a UML Diagram hyp

// Semantic (decomposition) rules
11;The goal model must be semantically correct;There exists at least a conn
12;Each goal or obstacle which decomposes through a complete AND cannot dec
13;Each goal or obstacle which decomposes through a partial AND cannot deco
14;Each goal or obstacle which decomposes through a complete OR can only de
15;Each goal or obstacle which decomposes through a partial OR can only dec
16;Each goal or obstacle which is obstructed by an obstacle or goal cannot
17;Each goal or obstacle which decomposes through a complete AND must decor
18;Each goal or obstacle which decomposes through a complete OR must decomp
19;Each goal or obstacle which decomposes through a complete AND must decor
20;Each goal or obstacle which decomposes through a partial AND must decomp
21;Each complete or partial AND which is connected via an AND anchor link t
22;Each complete or partial AND which is connected via an AND anchor link t
23;Each complete or partial AND which is connected via an AND anchor link t
24;Each complete or partial AND which is connected via an AND anchor link t

// Tree-structure rules
25;The goal decomposition graph must be a tree with the root node being a g
26;There must be exactly one root entity of the tree;The graph in the speci
27;The root entity of the tree must be a goal;The root entity of the tree t
28;Each non-root entity of the tree must have exactly one father;There exis
29;Each non-leaf entity of the tree must be either a goal, or an obstacle,
30;Each leaf entity of the tree must be an evidence;There exists at least a
31;The tree must have at least a leaf entity;The tree does not have any lea

// Evidence-Artifact multiplicity rules
32;Artifacts and evidences must support each other;There exists at least an
33;Each evidence must be supported by exactly one artifact;There exists at
34;Each artifact must support at least one evidence;There exists at least a

// Reference correctness rules
35;Each decomposition referenced by a diagram hyperlink must be semantical
36;Each goal or obstacle connected via a reference link to a diagram hyper
37;Each goal or obstacle connected via a reference link to a diagram hyper

// other rules
38;Each element must appear in at least one diagram;There exists at least o
39;Each diagram referenced by a diagram hyperlink must comply with all the
  
```

Figura 57: File Rules. csv contenente le definizioni delle regole di consistenza

3.4.4.2 Selezione di un goal model ed avvio della validazione

La validazione di un goal model viene avviata dal menu di Enterprise Architect "Project > Model Validation > Validate Selected" (Figura 58).

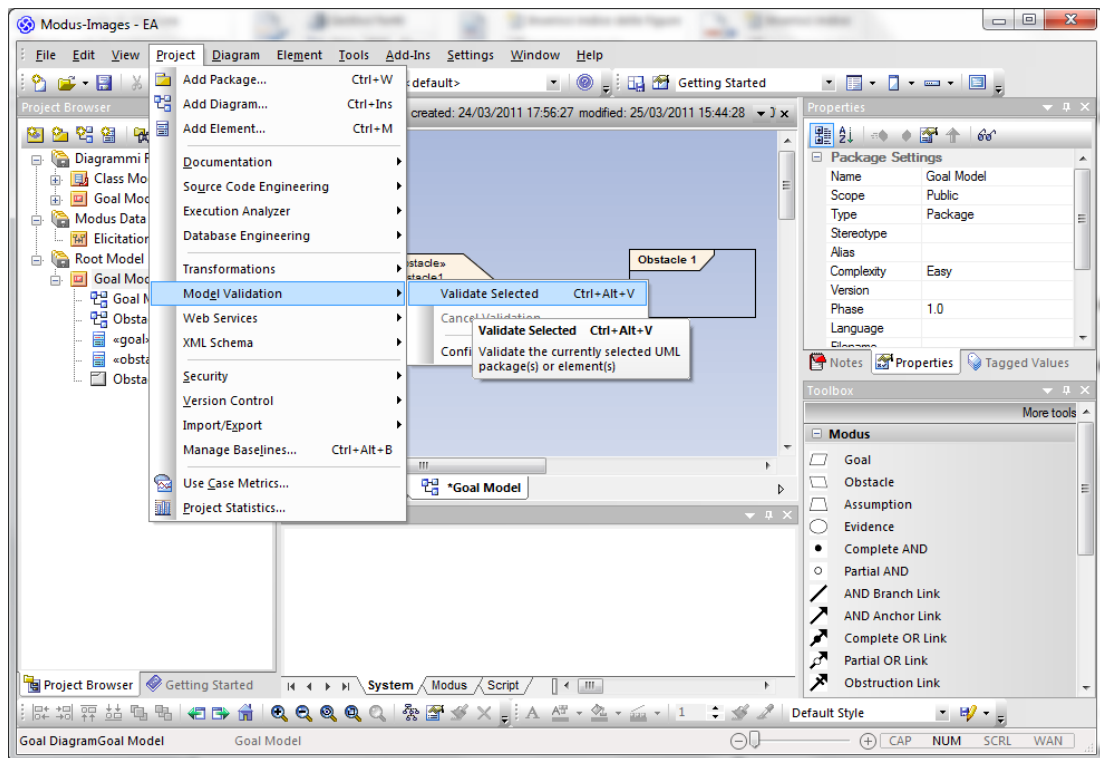


Figura 58: Avvio della validazione

Durante la validazione viene visualizzata una progress bar che notifica all'utente lo stato della stessa (Figura 59).

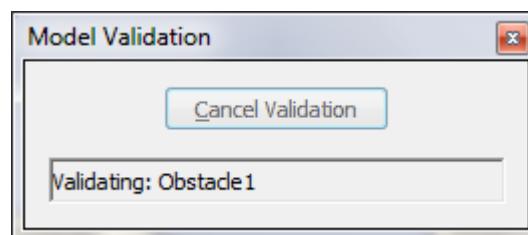


Figura 59: Status della validazione

3.4.4.3 Visualizzazione del report di validazione

Al termine della validazione, EA visualizza nella tab “Model Validation” della finestra Output il risultato della validazione (Figura 60).

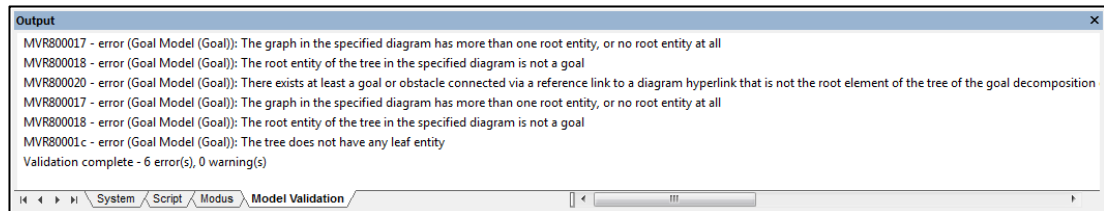


Figura 60: Report della validazione nella tab "Model Validation" di EA

Queste informazioni sono integrate con quelle, molto più complete e dettagliate, riportate nella tab “Modus” del tool (Figura 61).

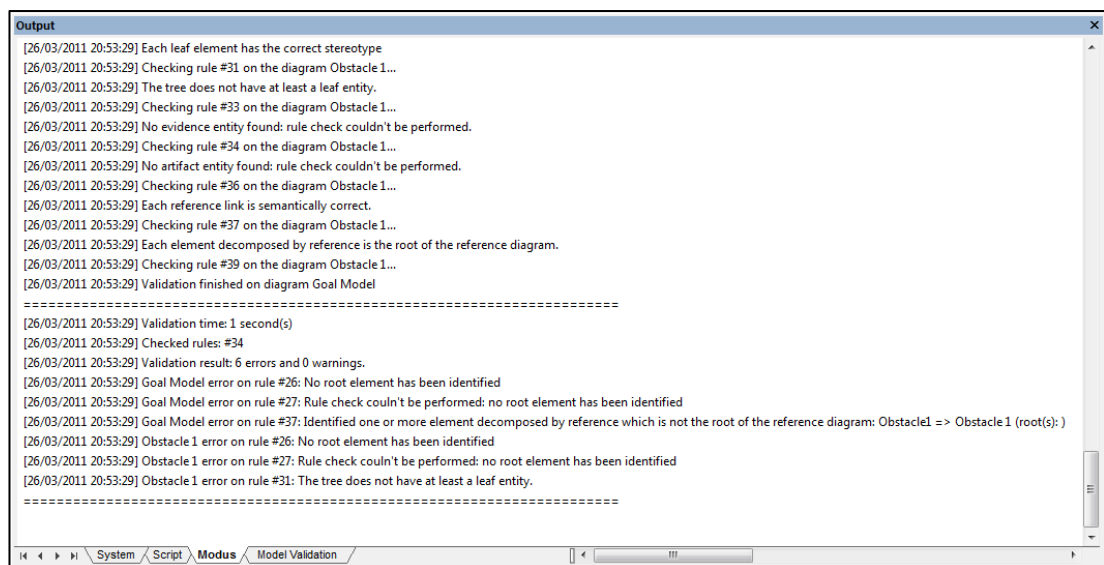


Figura 61: Report della validazione nella tab "Modus" di Modus Tool

3.5 Caso d'uso Elicitation: dalla definizione alla sua implementazione

3.5.1 Definizione

La feature di Elicitation consente all'utente di gestire sessioni di elicitazione dell'opinione di esperti in merito alle distribuzioni di probabilità di goal ed ostacoli foglia di un goal tree.

Con questa funzionalità, Modus Tool ha l'obiettivo di fornire all'utente uno strumento completo ed *user friendly* a supporto del task di elicitazione nella qualificazione delle nuove tecnologie secondo la metodologia Modus. Come ogni attività che si articola in più fasi e le cui informazioni sono rappresentate in più artefatti, l'elicitazione dell'opinione degli esperti trae considerevoli benefici dall'utilizzo di uno strumento di supporto informatico. La feature propone infatti un'organizzazione immediata ed intuitiva di tutte le risorse, umane e materiali, che prendono parte all'attività:

- Esperti ed altro personale (coordinatori e amministratori) sono modellati all'interno del programma, analogamente a quanto accade in qualunque applicazione i cui dati siano accessibili da più utenti contemporaneamente;
- La comunicazione tra le persone coinvolte è integrata nel tool, e resa possibile da una lavagna pubblica in cui tutti gli utenti possono lasciare messaggi;
- Elicitazioni contestuali di opinioni di diversi esperti in merito ad uno stesso goal tree sono logicamente organizzate in *sessioni*, aumentando il livello di astrazione dell'attività di Elicitation;
- Sessioni di elicitazione e risultati delle singole elicitazioni sono visualizzati simultaneamente nel dettaglio: un efficace layout a tabelle permette una visione esaustiva e d'insieme allo stesso tempo;

- La funzionalità di Elicitation è integrata all'attività di Goal Modeling, dovendo dipendere da essa: per ogni sessione è possibile visualizzare tutti i diagrammi di riferimento al momento dell'ultimo salvataggio, in modo da consentire all'utente di accorgersi di eventuali cambiamenti del modello durante l'attività di elicitazione. Inoltre, le informazioni principali riguardo le entità su cui verte l'elicitazione sono riportate sia nella finestra principale del tool che nella finestra del questionario di elicitazione, evitando all'utente l'inconvenienza di dover visualizzare alternativamente il goal model ed il riepilogo dell'attività di elicitazione;
- Le risposte degli esperti ai questionari di elicitazione sono mantenute riservate per evitare di influenzare le opinioni, e sono comunque salvate solo dopo l'esplicita conferma dell'esperto, fornendo la possibilità agli esperti di correggere eventuali errori ed esprimere la loro opinione a più riprese;
- Il salvataggio del lavoro svolto avviene garantendo comunque all'utente la possibilità di mantenere una *history* del progresso del lavoro attraverso backup successivi, utile a monitorare l'attività nel corso del tempo;
- L'*export* dei dati delle elicitazioni di una sessione, che prenderanno successivamente parte a simulazioni con metodo Montecarlo, garantisce flessibilità in fase di gestione del *bias* delle opinioni degli esperti: è possibile aggregare i risultati delle elicitazioni di più esperti che hanno preso parte alla sessione, trascurando i dati di alcuni di essi.

3.5.2 Activity diagrams: specifica delle operazioni

L'utente accede alla funzionalità di Elicitation dalla relativa voce nel menu dell'add-in. Il *workflow* di un'attività di elicitazione delle opinioni degli esperti inizia con la creazione di un goal model, procedendo con la compilazione ed il salvataggio dei questionari e terminando con l'export dei dati (Figura 62).

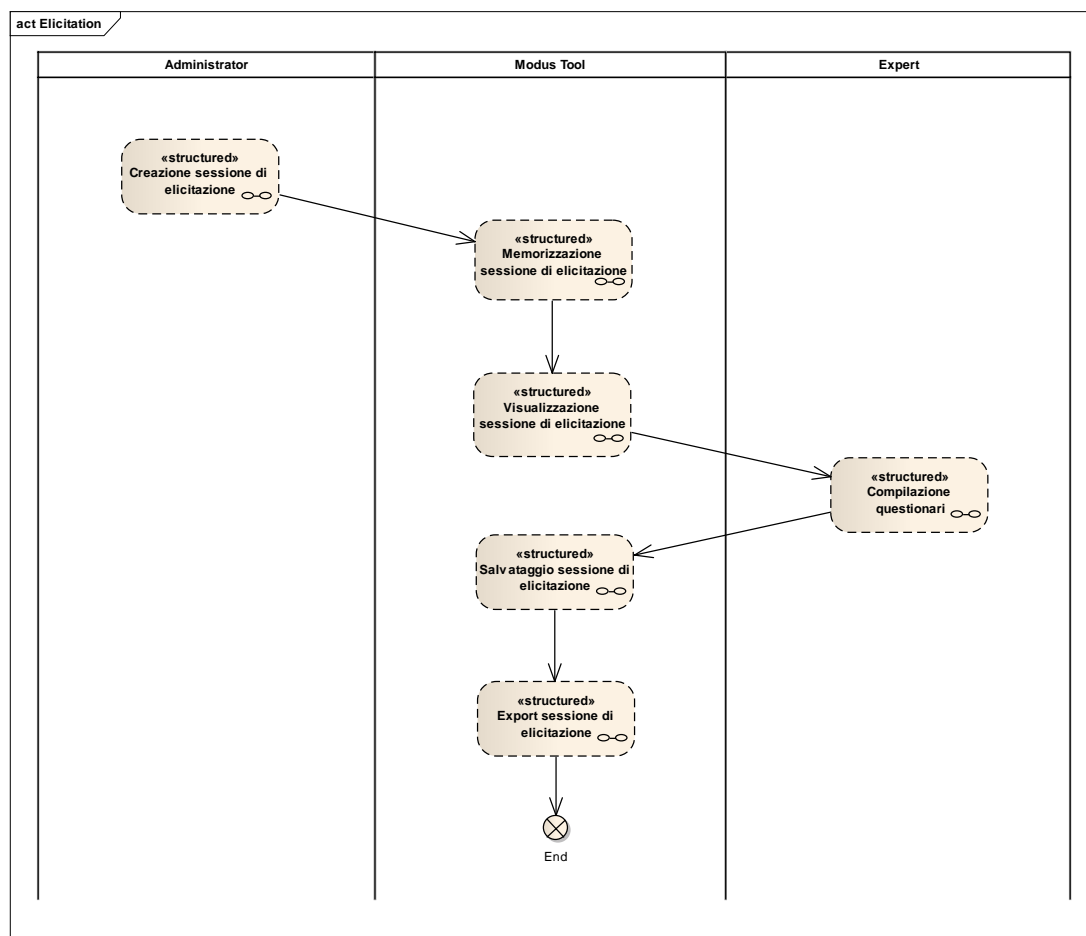


Figura 62: Activity diagram della feature di Elicitation

Si assume che l'utente sia identificato nel sistema, prima di effettuare ognuna delle operazioni della feature di Elicitation.

3.5.2.1 Creazione sessione di elicitazione

La Figura 63 riporta il dettaglio della structured activity relativa alla creazione (lato utente) di una sessione di elicitazione.

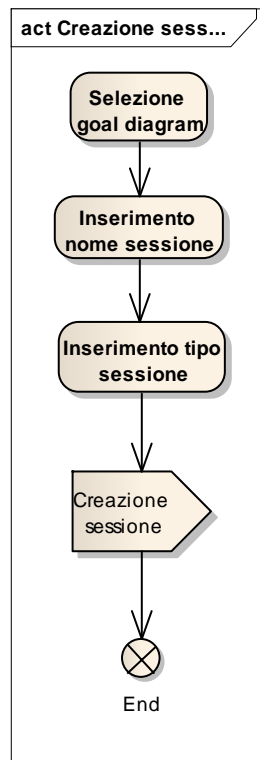


Figura 63: Structured activity "Creazione sessione di elicitazione"

Accedendo all'apposita funzionalità del tool, l'utente definisce il goal diagram principale, il nome ed il tipo della sessione (standard o avanzata), confermando l'operazione di creazione.

3.5.2.2 Memorizzazione sessione di elicitazione

La Figura 64 riporta il dettaglio della structured activity relativa alla creazione e alla memorizzazione (lato tool) di una sessione di elicitazione.

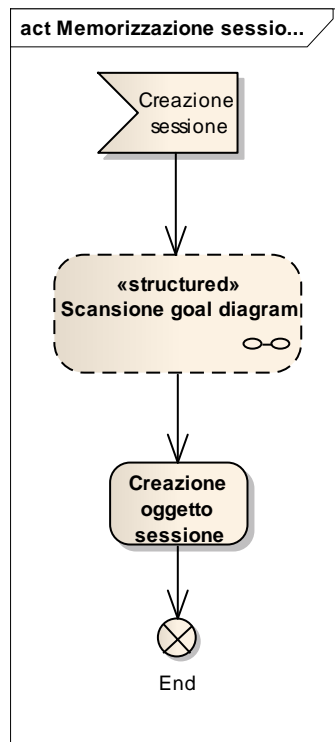


Figura 64: Structured activity "Memorizzazione sessione di elicitazione"

La memorizzazione di una sessione prevede in un primo momento la scansione del goal diagram principale della sessione (Sezione 3.5.2.2.1). Successivamente le informazioni statiche riguardo la sessione creata (nome, tipo, autore e data di creazione), vengono raccolte e memorizzate all'interno di un oggetto nel repository di Enterprise Architect.

3.5.2.2.1 Scansione goal diagram

La Figura 65 riporta il dettaglio della structured activity relativa alla scansione del goal diagram principale di una sessione durante l'attività di memorizzazione di una sessione.

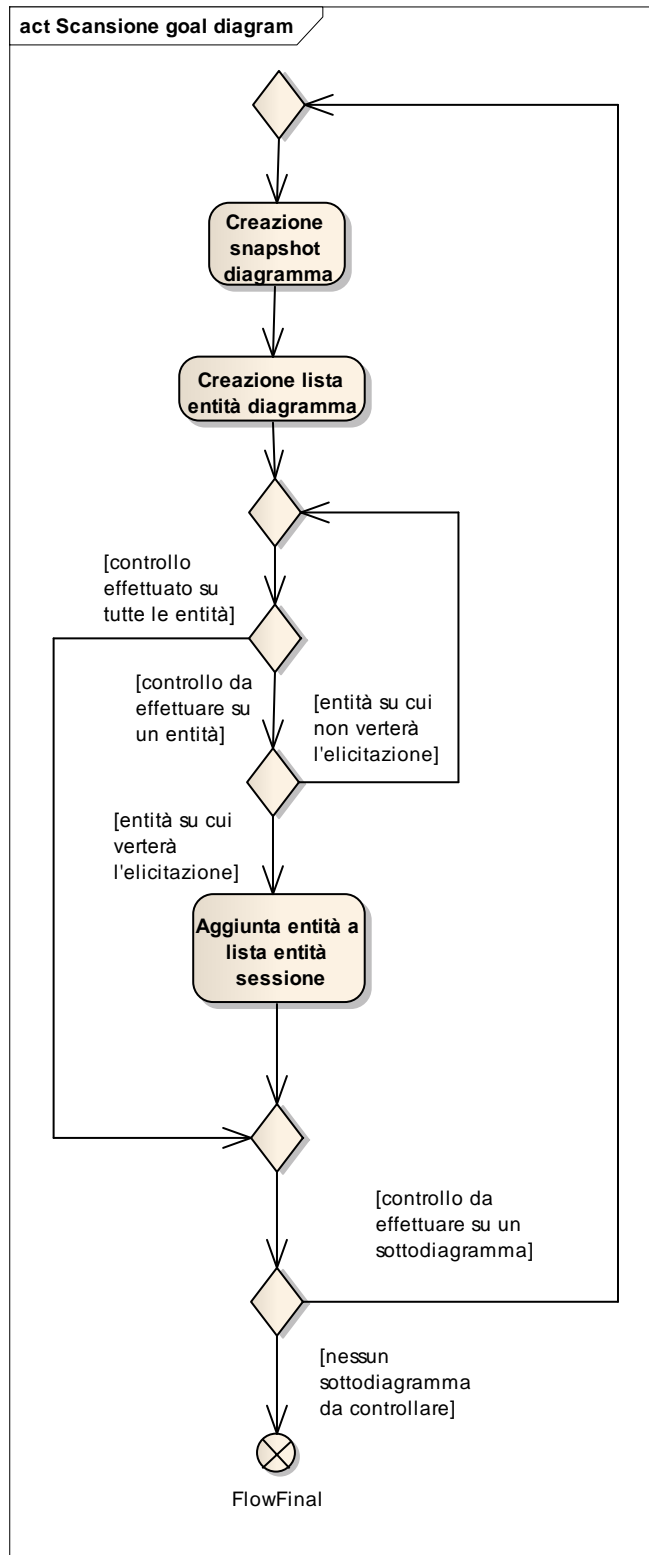


Figura 65: Structured activity "Scansione goal diagram"

La scansione di un goal diagram è riconducibile ad un semplice algoritmo di esplorazione di un albero definito in più livelli, ognuno dei quali ne identifica uno o più sottoalberi (Sezione 3.3.1.1).

Per ogni diagramma dell'albero dei diagrammi di decomposizione, vengono salvate delle snapshot che ne fotografano la rappresentazione attuale. Successivamente le entità in esso contenute sono inserite in una lista di controllo. Queste sono quindi esaminate ed aggiunte nella lista delle entità della sessione nel caso in cui sia necessario elicitare la distribuzione di probabilità relativa, cioè nel caso di goal o ostacoli foglia, oppure di AND incompleti per goal ed OR incompleti per ostacoli.

3.5.2.3 Visualizzazione sessione di elicitazione

La Figura 66 riporta il dettaglio della structured activity relativa alla visualizzazione dei dati di una sessione di elicitazione.

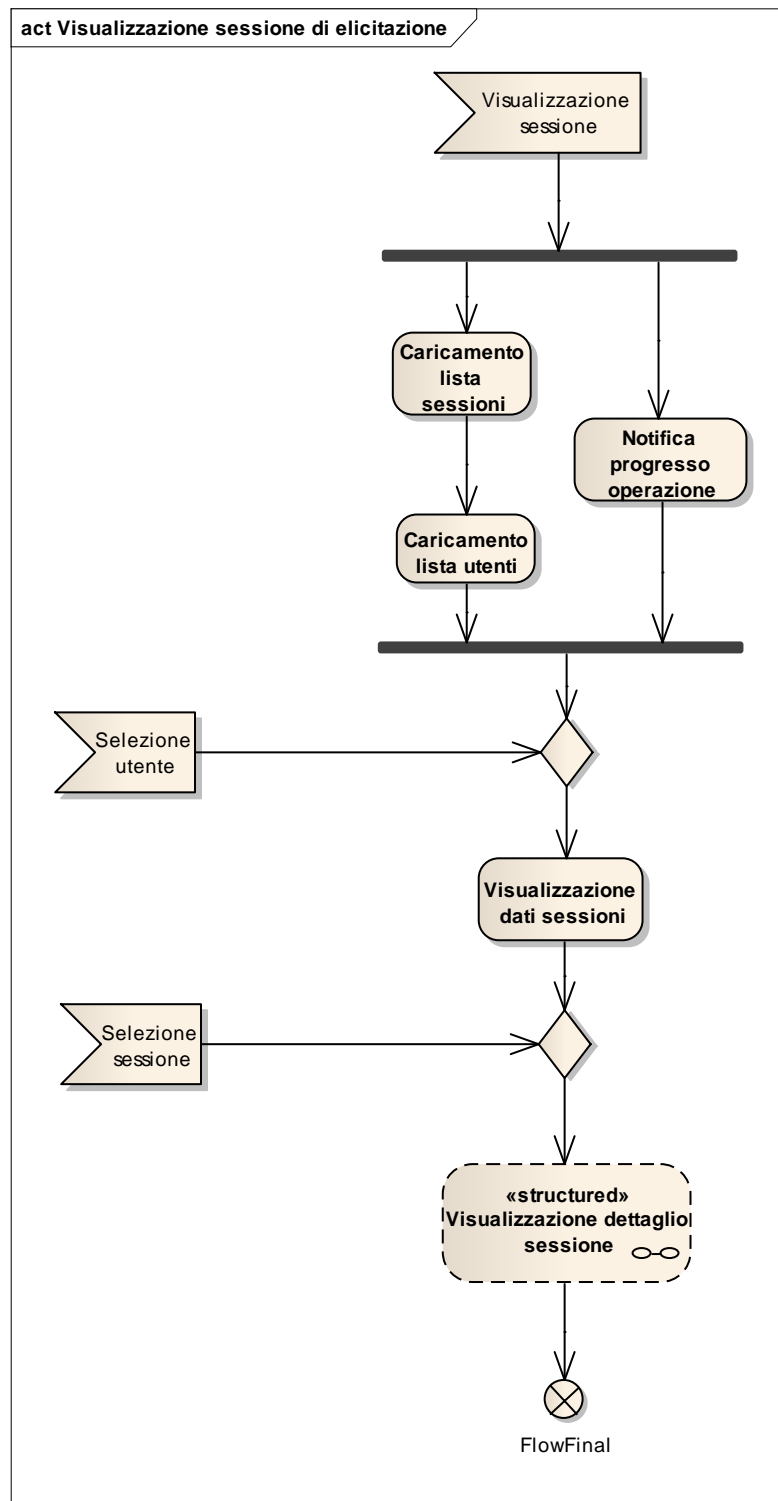


Figura 66: Structured activity "Visualizzazione sessione di elicitazione"

Quanto l'utente attiva il caso d'uso di visualizzazione di una sessione (Sezione 3.5.2.3), il sistema carica la lista delle sessioni e la lista degli utenti, notificando

all'utente lo status dell'operazione. Successivamente l'utente si identifica all'interno del sistema, che provvede ad aggiornare i dati delle sessioni con le informazioni proprie dell'utente riguardo la sua eventuale partecipazione alla sessione.

La selezione di una sessione dalla lista attiva l'operazione di visualizzazione del suo dettaglio, specificata nella Sezione 3.5.2.3.1.

3.5.2.3.1 Visualizzazione dettaglio sessione

La Figura 67 riporta il dettaglio della structured activity relativa alla visualizzazione dei dati di una sessione di elicitazione.

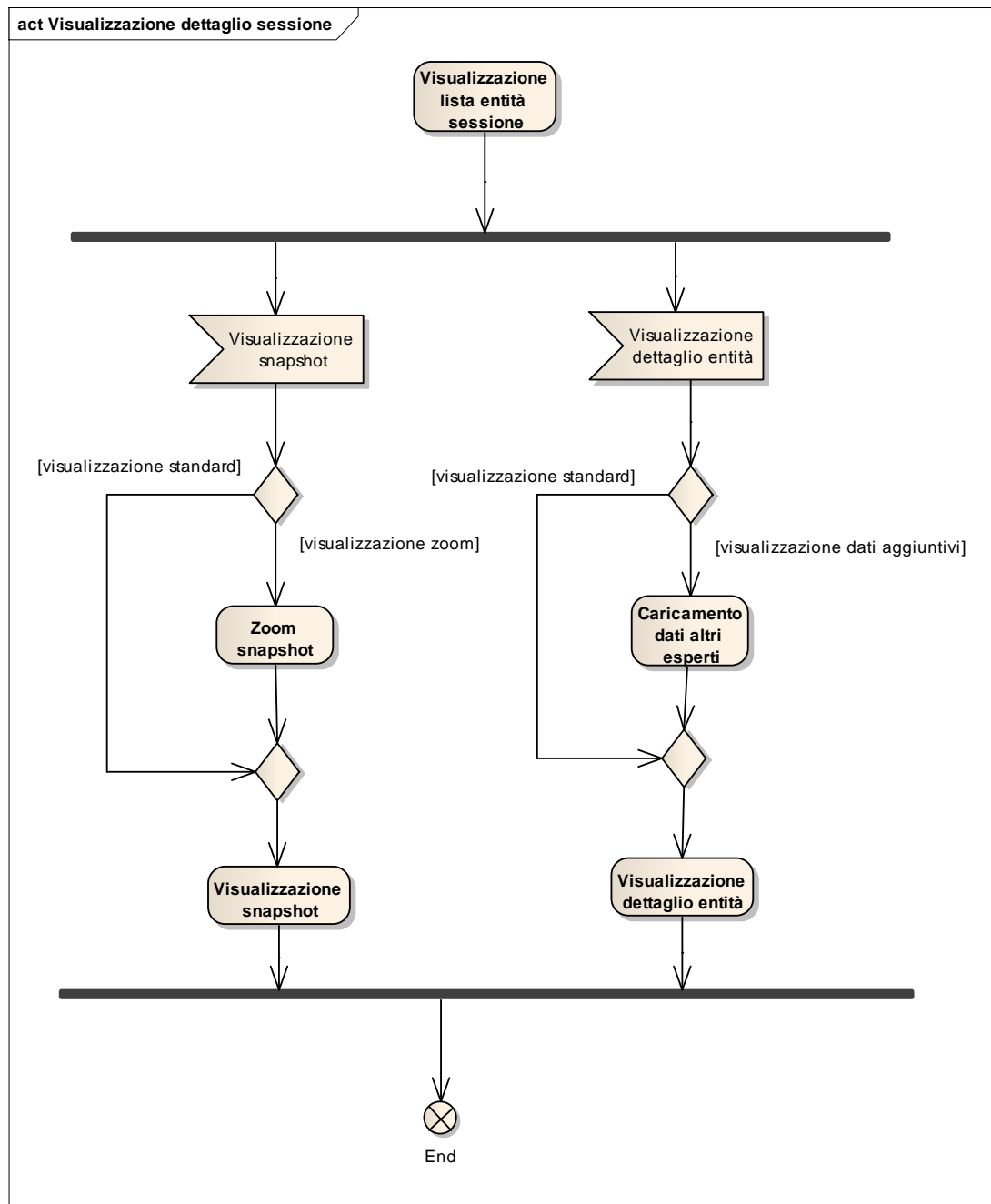


Figura 67: Structured activity "Visualizzazione dettaglio sessione"

Il sistema visualizza la lista delle entità contenute (anche in eventuali diagrammi di decomposizione per riferimento) nel diagramma principale della sessione. Successivamente l'utente può scegliere, in qualsiasi ordine, di visualizzare le snapshot dei diagrammi o il dettaglio di un'entità della sessione. Qualora l'utente lo specifichi esplicitamente, il sistema visualizza uno zoom della snapshot, o arricchisce

il dettaglio di un'entità con i dati delle elicitazioni delle distribuzioni degli altri esperti.

3.5.2.4 Compilazione questionari

La Figura 68 riporta il dettaglio della structured activity relativa alla compilazione da parte di un esperto di un questionario di elicitazione.

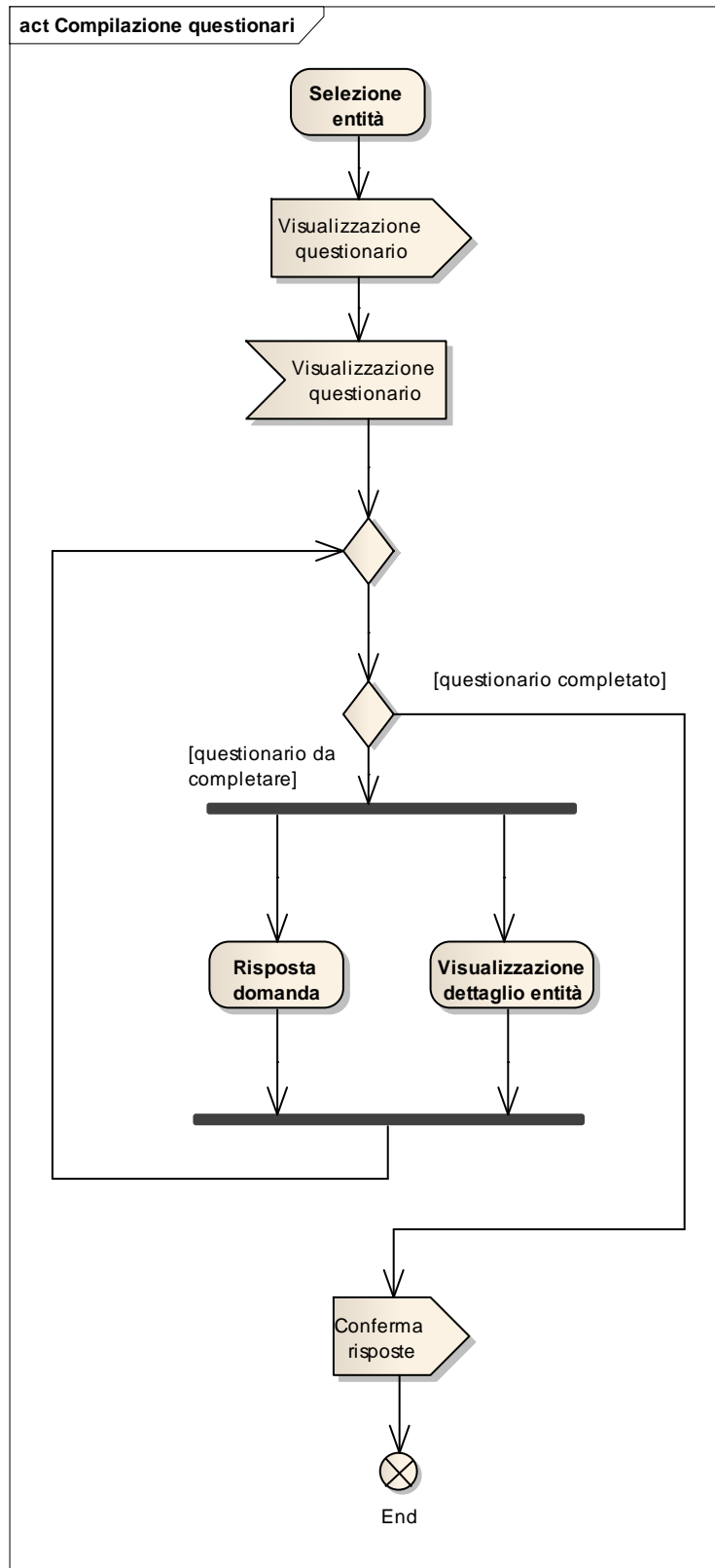


Figura 68: Structured activity "Compilazione questionari"

L'utente seleziona un'entità contenuta in un diagramma della sessione, e successivamente visualizza il questionario relativo all'entità presentato dal sistema. L'utente può visualizzare informazioni dettagliate sull'entità contestualmente all'inserimento delle risposte alle domande. Al termine dell'operazione, l'utente conferma le risposte ed il sistema memorizza temporaneamente i dati inseriti.

3.5.2.5 Salvataggio sessione di elicitazione

La Figura 69 riporta il dettaglio della structured activity relativa al salvataggio permanente delle modifiche apportate ad una sessione.

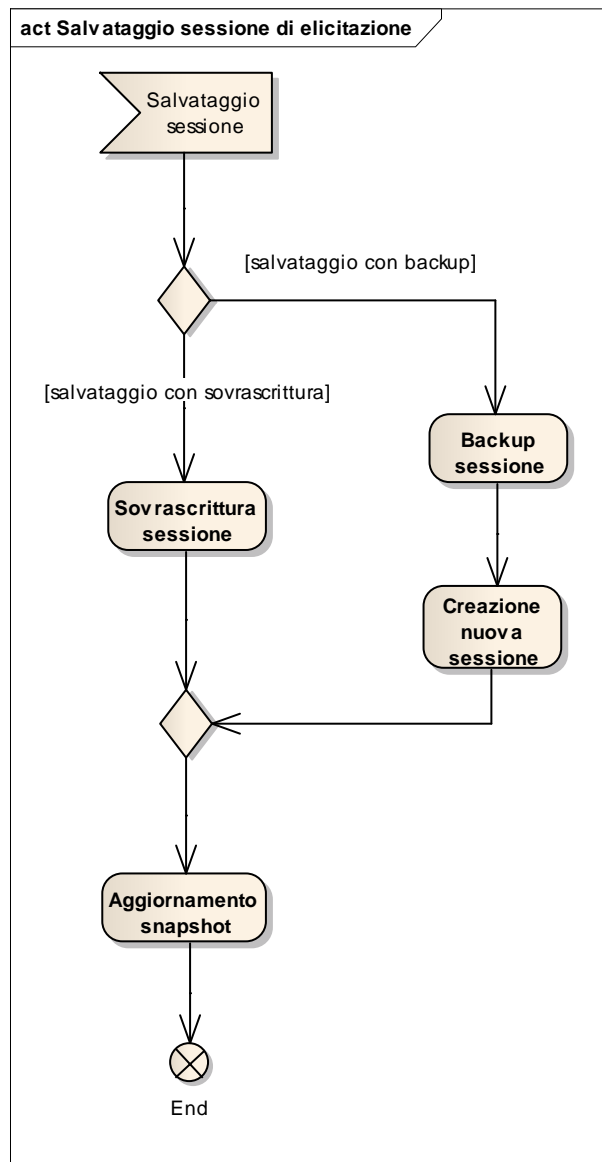


Figura 69: Structured activity "Salvataggio sessione di elicitazione"

Una sessione di elicitazione può essere completata a più riprese: al termine di un'attività lavorativa è possibile salvare permanentemente le modifiche apportate ad una sessione. Il salvataggio avviene secondo le già discusse modalità di sovrascrittura e backup, in alternativa tra loro. Al momento del salvataggio, il sistema ricrea le snapshot dei diagrammi della sessione: ciò impone un *reset* dello stato di consistenza di una sessione. Si assume pertanto che il salvataggio di una sessione inconsistente abbia la semantica di una validazione della stessa da parte dell'utente, considerando la modifica di un diagramma successiva alla creazione

della sessione non influente ai fini del modello. Questa eventualità accade spesso quando vengono effettuati *cosmetic changes* al diagramma come ad esempio piccole variazioni della posizione di elementi o connettori. Al contrario, una sessione che presentasse inconsistenze fondamentali non dovrebbe essere validata utilizzando l'opzione di salvataggio al fine di evitare di ottenere dati di elicitazione che non siano effettivamente rispondenti ad un goal model reale.

3.5.2.6 Export sessione di elicitazione

La Figura 70 riporta il dettaglio della structured activity relativa all'export dei dati di elicitazione di una sessione.

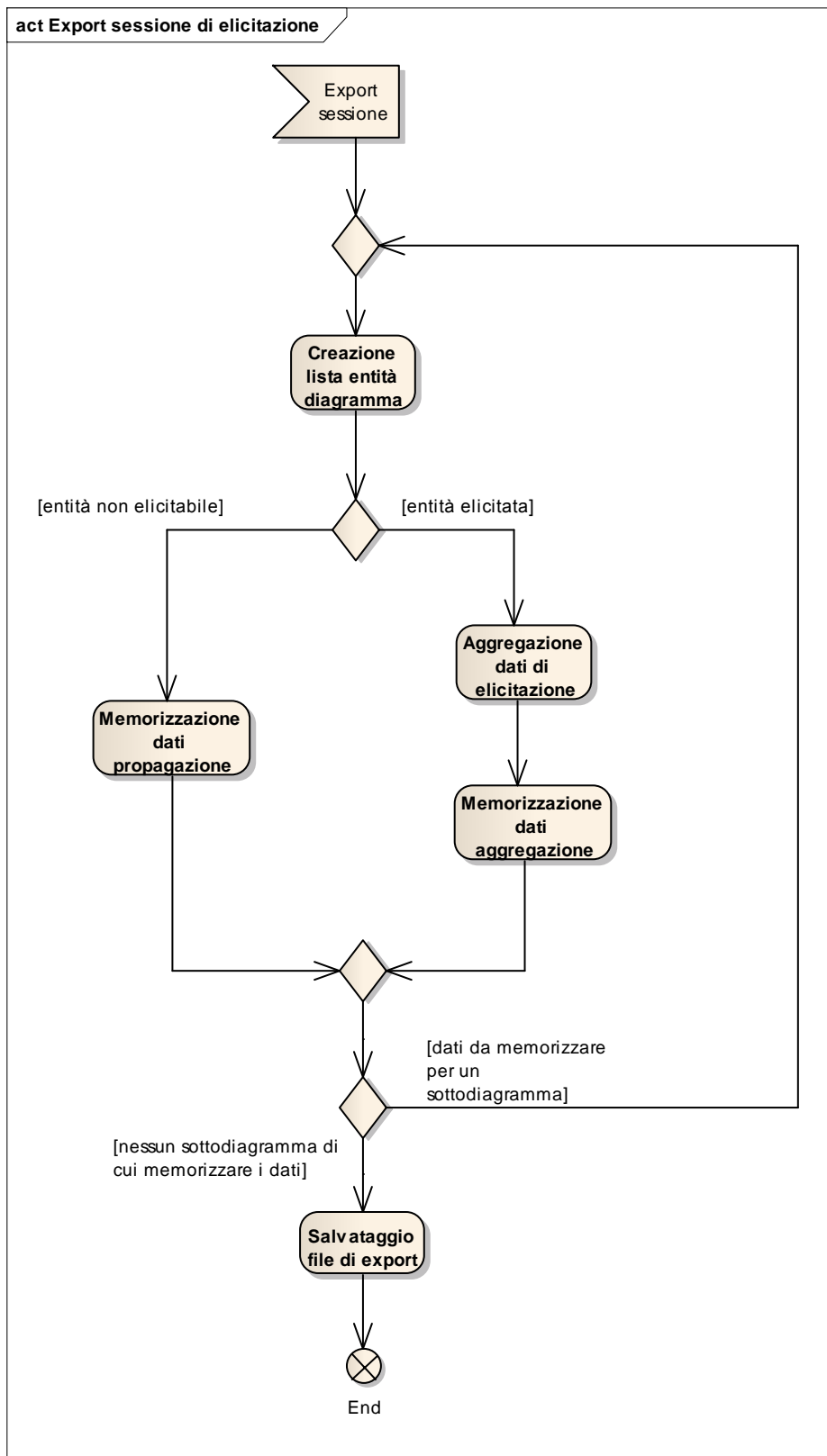


Figura 70: Structured activity "Export sessione di elicitazione"

Una volta completata l'attività di elicitazione, i dati forniti dagli esperti devono essere raccolti e propagati dalle entità foglia a tutte le altre entità del goal tree (Sezione 2.7). Sono pertanto analizzati il diagramma principale della sessione e tutti i suoi sottodiagrammi:

- Per ogni entità elicitata viene memorizzata la distribuzione aggregata di tutti gli esperti di cui l'utente ha specificato di esportare i dati;
- Per ogni altra entità, vengono memorizzati i dati di propagazione delle distribuzioni .

Al termine dell'analisi del goal tree, i dati memorizzati sono salvati all'interno di un file nel path specificato dall'utente.

3.5.3 Progettazione: analisi del problema e scelte architettureali

La feature di Elicitation è stata realizzata come applicativo *stand-alone* accessibile dal menu di Enterprise Architect.

È stato pertanto necessaria un'attenta attività di progettazione, al fine di consentire la realizzazione di un sistema che fosse flessibile, estendibile e manutibile. Il supporto fornito all'attività di elicitazione verte su quattro aspetti principali:

- Modellazione degli esperti;
- Realizzazione di una struttura di elicitazione e riepilogo dei dati forniti dagli esperti;
- Memorizzazione delle informazioni all'interno del modello di Enterprise Architect;
- Presentazione delle informazioni all'utente.

Particolare attenzione è stato pertanto posta nella modellazione del processo di *opinions elicitation* (Sezione 3.5.2), le cui astrazioni principali sono rappresentate dai concetti di progetto, sessione, entità ed elicitazione (Sezione 3.2.1).

La modellazione degli esperti è l'unica feature della funzionalità di Elicitation che non è stata concepita *ex-novo*, essendo essa stata completamente realizzata sfruttando la funzionalità nativa di Enterprise Architect per la gestione del personale coinvolto in un progetto.

È stato invece definito un modello dei dati di elicitazione, riportato in Figura 71.

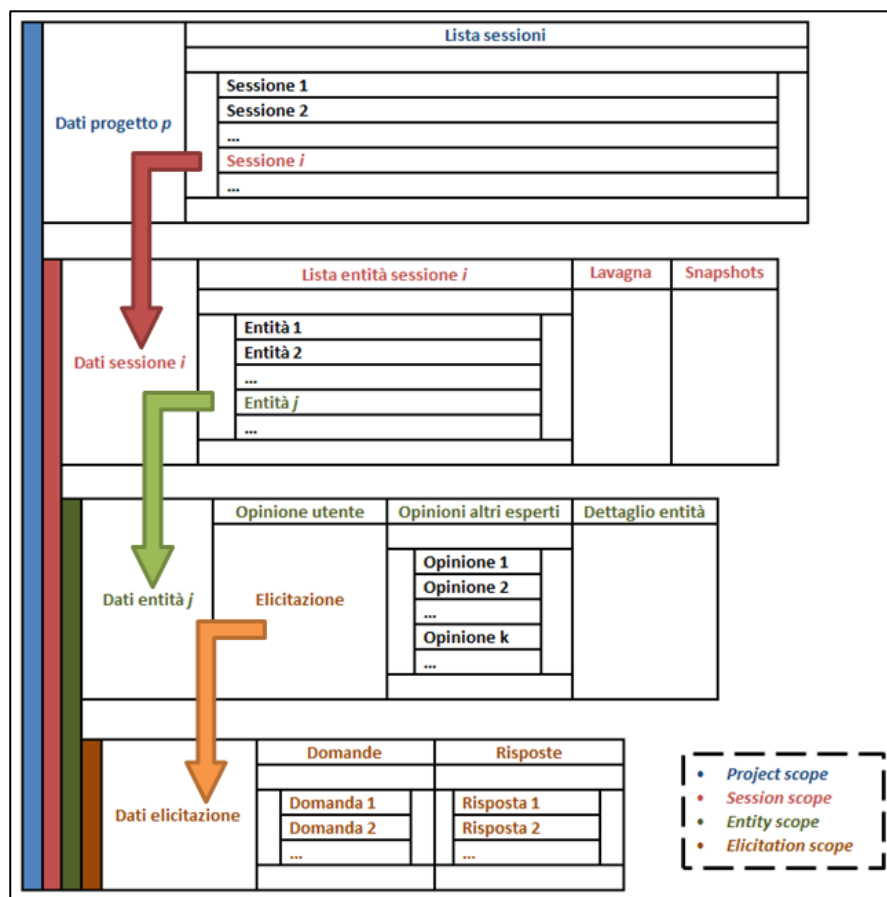


Figura 71: Modello dei dati della feature di Elicitation

Ogni progetto ha una lista di sessioni (esistenti nel *project scope*), le quali contengono a lista delle entità del goal tree relativo alla sessione stessa (esistenti nel *session scope*). Ogni entità rappresenta un evento sul quale gli esperti esprimono opinioni (esistenti nell'*entity scope*) rispondendo ad un questionario di elicitazione (esistente nell'*elicitation scope*). La gerarchia di contenimento degli scope fa sì che modifiche apportate in uno scope interno siano visibili all'esterno unicamente dopo il *commit* delle stesse, che le rende persistenti. Si osserva come la

separazione degli ambiti in cui esistono i dati aumenta notevolmente il grado di flessibilità della gestione dei dati:

- La validazione sintattica delle risposte ad un questionario, avviene in tempo reale nell'elicitation scope: non è possibile confermare l'inserimento di risposte i cui dati siano errati, rappresentando ad esempio valori di probabilità negativi o maggiori di 1. Questo approccio garantisce pertanto l'atomicità di un'operazione di elicitazione della distribuzione di probabilità di un singolo evento, e la correttezza sintattica dei dati forniti dall'esperto;
- Il salvataggio temporaneo delle risposte relative ad un questionario avviene alla conferma delle risposte di un questionario nell'entity scope. Data la complessità della qualificazione di nuove tecnologie, è bene che le risposte ai questionari possano essere modificate in momenti successivi al loro primo inserimento, consentendo all'esperto di correggere eventuali errori di valutazione commessi in prima battuta;
- L'insieme delle opinioni espresse da un esperto relative alle entità di una sessione, così come le informazioni condivise nella lavagna vivono nel session scope. Il session scope ha carattere transitorio, ed è l'unico livello logico di organizzazione dei dati (ad esclusione del project scope, che è però gestito da Enterprise Architect) in cui è possibile rendere i cambiamenti persistenti;
- Tutti i dati relativi ad un'attività di qualificazione, vivono nel project scope, che include pertanto le opinioni di tutti gli esperti coinvolti. La gestione del project scope, implementata attraverso un DBMS interno di EA che lavora su repository memorizzati su disco, è completamente delegata all'Automation Interface.

Il meccanismo degli scope dei dati supporta appieno la natura asincrona ed iterativa del processo di qualificazione, fornendo uno strumento che non richieda la presenza simultanea di tutti gli esperti e consenta la rivisitazione del lavoro svolto.

La persistenza dei dati relativi alle sessioni di elicitazione è, come già accennato, affidata al repository di EA: in tal modo tutti i dati sono contenuti all'interno del file EAP contenente le informazioni di un progetto. Questa scelta consente di condividere facilmente un progetto di qualificazione tra diversi sistemi.

Particolare attenzione è stata infine posta allo sviluppo dell'interfaccia grafica. In ogni sistema informativo lo strato di presentazione deve rispettare elevati standard qualitativi, e Modus Tool non fa eccezione. Le informazioni su sessioni ed elicitazioni sono mantenute organicamente in un layout a tabelle, in cui è possibile navigare attraverso gli scope attraverso la semplice selezione di oggetti. Il riepilogo di una sessione riporta informazioni testuali e grafiche sul goal tree di riferimento, compattando tutte le informazioni necessarie all'esperto in fase di elicitazione all'interno di un'unica finestra. Lo stato di avanzamento di operazioni computazionalmente onerose come caricamenti, salvataggi ed export di dati è monitorato attraverso l'utilizzo di *progressbar*, come è pratica standard in casi analoghi.

3.5.4 Screenshots

Si presenta infine una verifica di funzionamento della feature di Elicitation, corredata da screenshot relativi agli step del flusso base della funzionalità che è stato presentato in forma di activity diagram nella Sezione 3.5.2.

3.5.4.1 Avvio della feature e finestra principale

È possibile avviare la finestra principale della feature di Elicitation di Modus Tool dal menu "Add-in > Modus > Elicit Probabilities" (Figura 72).

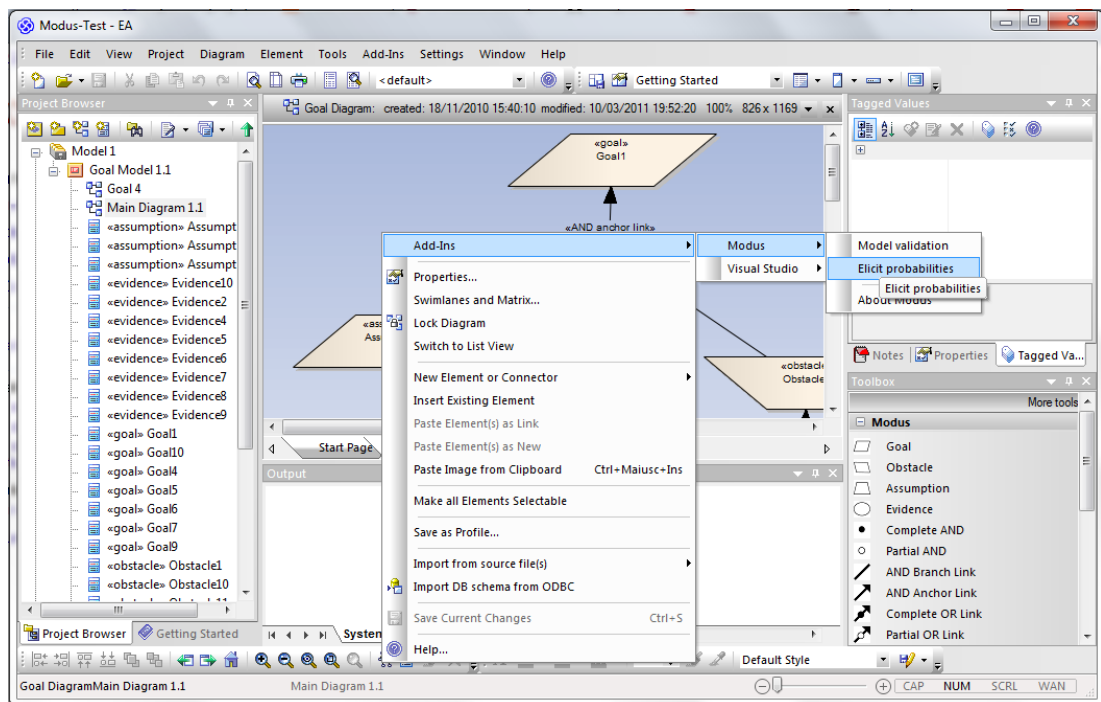


Figura 72: Avvio della feature di Elicitation dell'add-in

All'avvio della finestra, visualizza in alto a destra la lista degli utenti e precarica la lista delle sessioni, notificando il progresso dell'operazione nella status bar in basso (Figura 73).

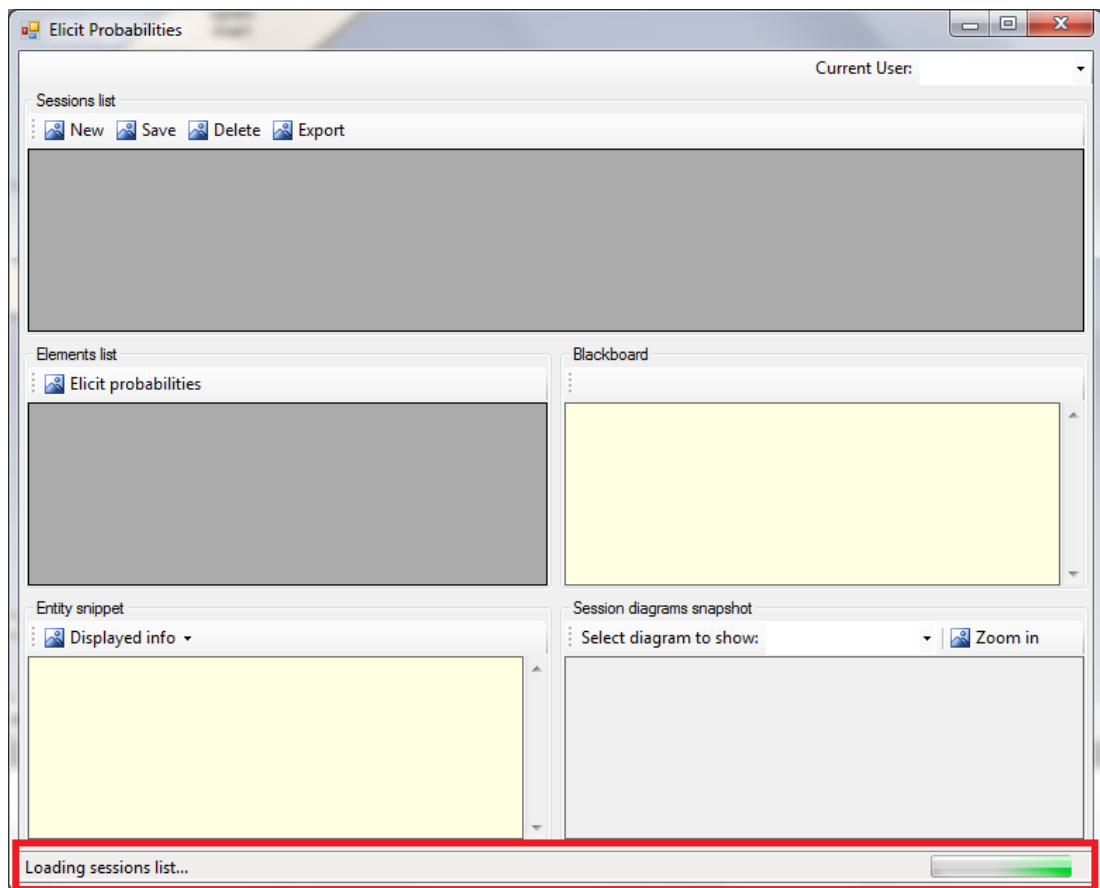


Figura 73: Finestra principale della feature di Elicitation

Alla selezione di un utente, il sistema visualizza nel pannello superiore la lista delle sessioni caricate con le loro informazioni di base (Figura 74).

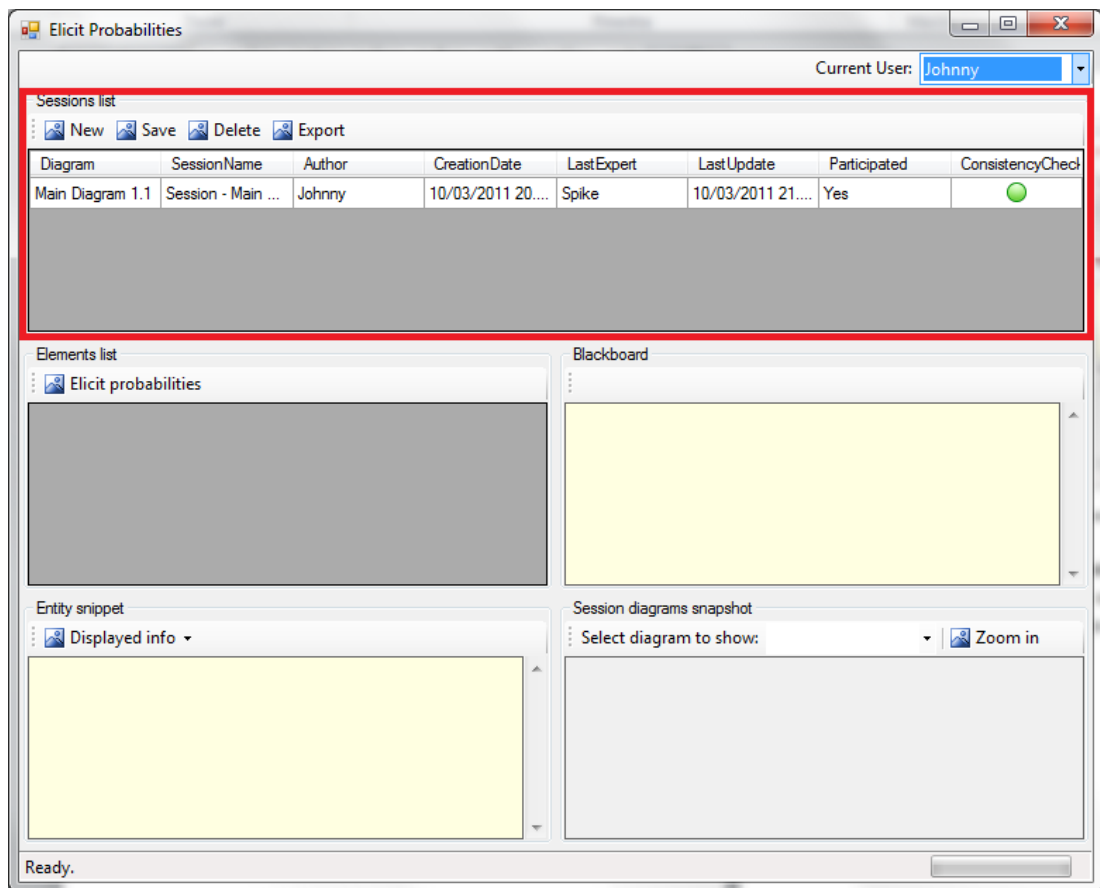


Figura 74: Accesso utente e lista delle sessioni

3.5.4.2 Creazione di una sessione

Cliccando sul pulsante “New” nella barra superiore, l’utente visualizza la finestra di creazione della sessione (Figura 75).

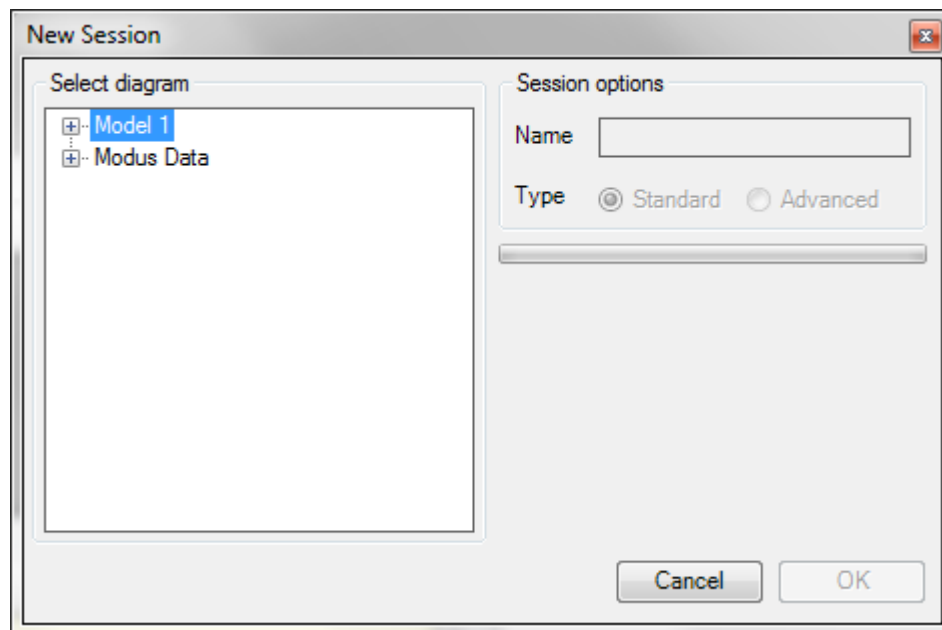


Figura 75: Finestra di creazione di una sessione (1)

Il bottone “OK” di conferma diventa attivo solo quando l’utente ha inserito correttamente tutti i parametri di creazione: diagramma principale, dal project browser sulla sinistra, e nome e tipo, dal pannello sulla destra (Figura 76).

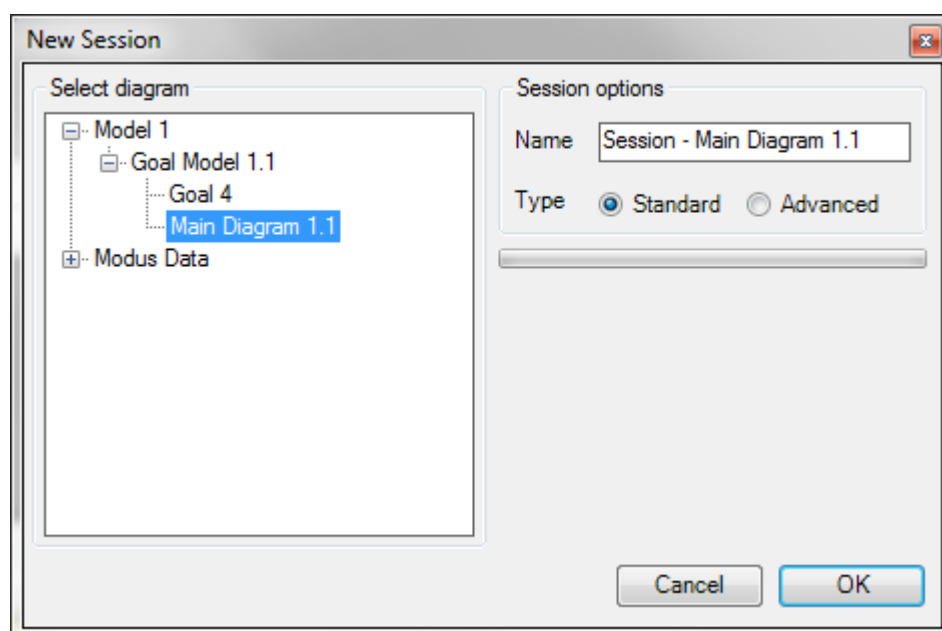


Figura 76: Finestra di creazione di una sessione (2)

3.5.4.3 Selezione di una sessione

Al termine della creazione, la sessione viene visualizzata nella tabella del pannello superiore della finestra principale. Selezionando la sessione, l'utente visualizza nei pannelli centrali la lista delle entità della sessione nella parte sinistra e la lavagna del progetto nella parte destra (Figura 77).

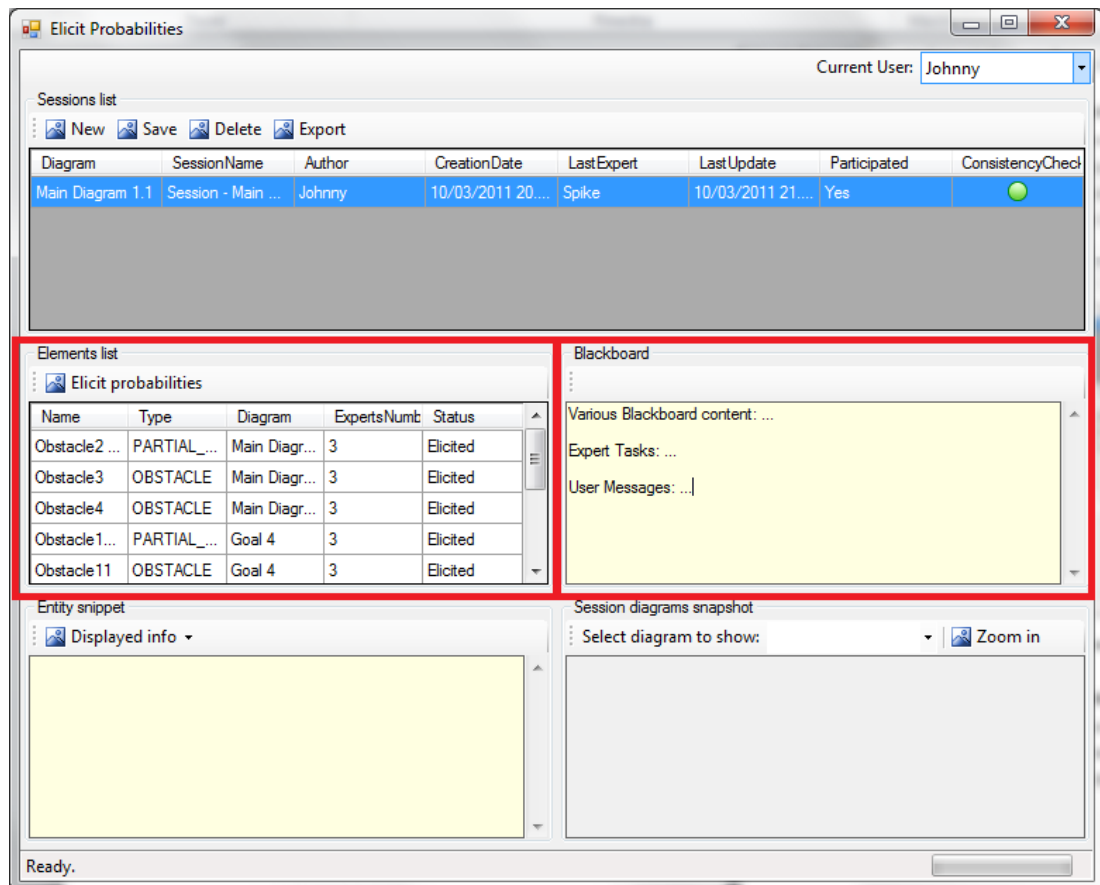


Figura 77: Entità di una sessione e lavagna

3.5.4.4 Visualizzazione di una snapshot di un diagramma

Selezionando un diagramma nella lista del pannello in basso a destra, l'utente visualizza in un riquadro compatto l'immagine del diagramma al momento dell'ultimo salvataggio della sessione (Figura 78).

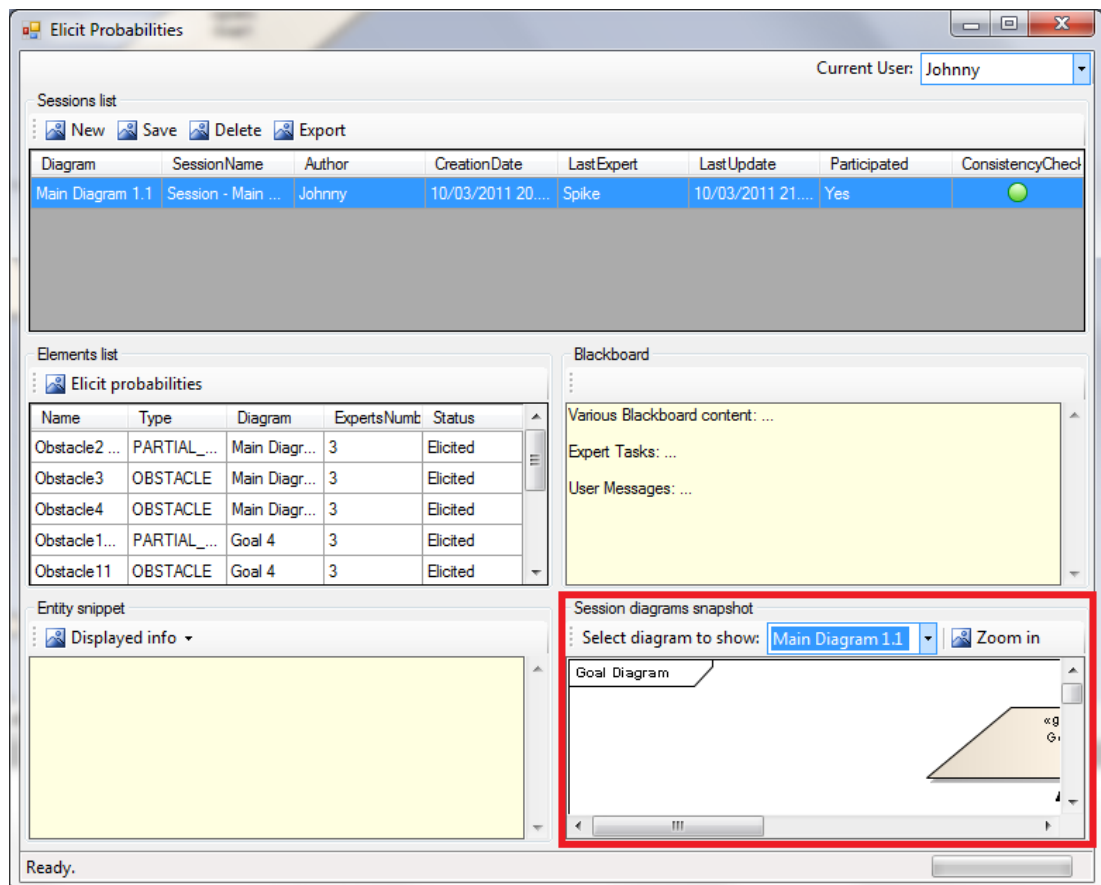


Figura 78: Snapshot di un diagramma della sessione

Cliccando su “Zoom in”, l’utente visualizza una finestra estesa ridimensionabile e scrollabile con l’immagine (Figura 79).

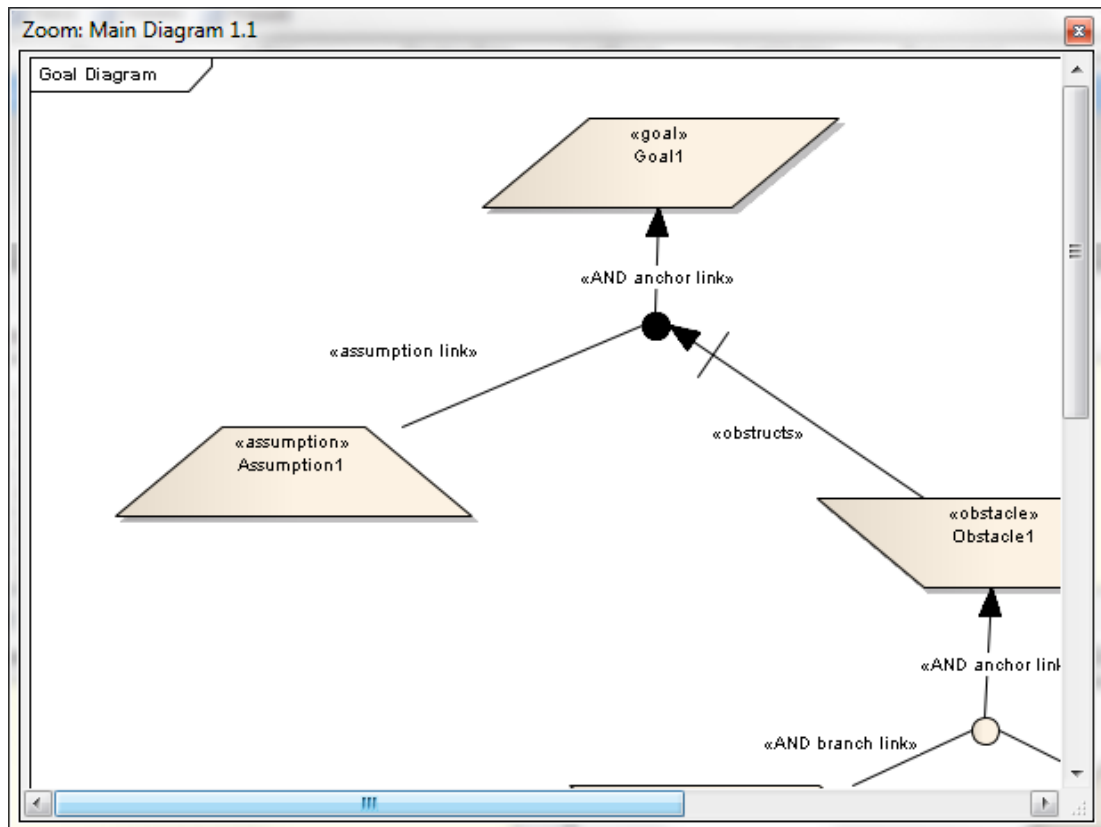


Figura 79: Finestra di zoom di una snapshot

3.5.4.5 Visualizzazione del dettaglio di un'entità del diagramma

Selezionando una sessione nel pannello centrale di sinistra, l'utente visualizza nel pannello in basso a sinistra le informazioni dettagliate relative all'entità selezionata (Figura 80).

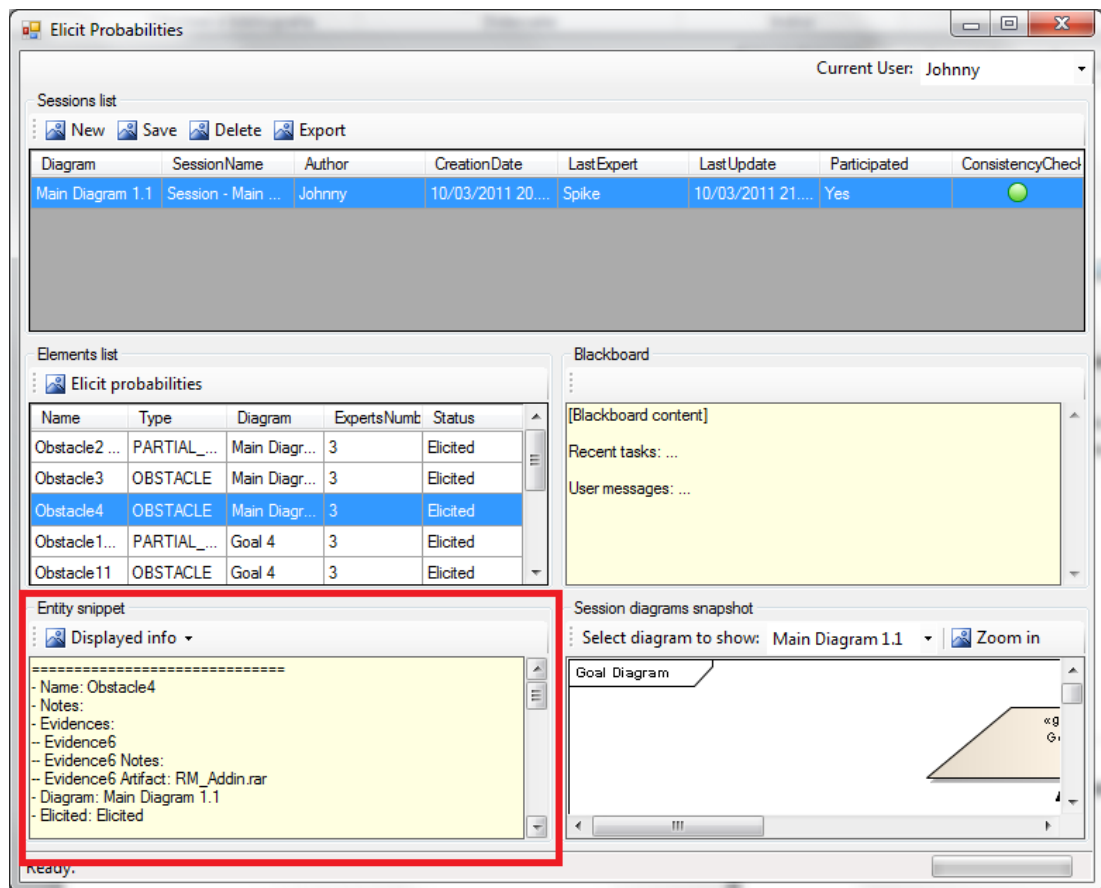


Figura 80: Informazioni sull'entità selezionata (1)

Selezionando l'opzione "Show other experts data" è possibile visualizzare le opinioni espresse dagli altri esperti (Figura 81).

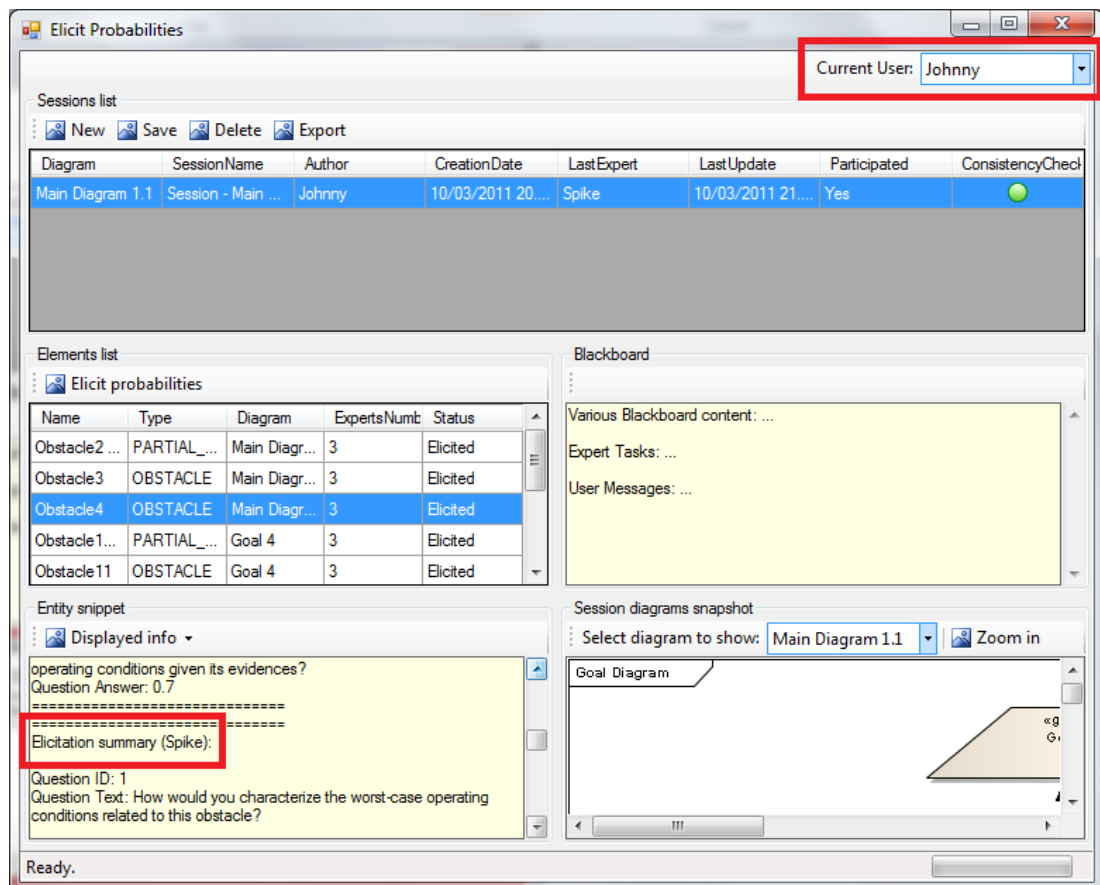


Figura 81: Informazioni sull'entità selezionata (2)

3.5.4.6 Elicitazione di un'opinione

Dopo aver selezionato un'entità nella lista, l'utente accede alla finestra del questionario di elicitazione cliccando su "Elicit opinions", visualizzando nel pannello di sinistra la lista delle domande del questionario e sulla destra il dettaglio sull'entità già visualizzato nella finestra principale (Figura 82).

The window titled "Entity Elicitation: Obstacle4" contains a "Questionnaire" table and an "Entity Snippet" panel.

Id	Text	Answer
1	How would you characterize the worst-case operating conditions related to this obstacle?	
2	How likely is this obstacle to occur in the worst-case operating conditions given its evidences?	
3	How would you characterize the best-case operating conditions related to this obstacle?	
4	How likely is this obstacle to occur in the best-case operating conditions given its evidences?	
5	How would you characterize the average-case operating conditions related to this obstacle?	

The "Entity Snippet" panel on the right displays the following information:

- Name: Obstacle4
- Notes:
- Evidences:
- Evidence6
- Evidence6 Notes:
- Evidence6 Artifact: RM_Addin.rar
- Diagram: Main Diagram 1.1
- Elicited: Elicited
- Elicited by 3 expert(s): Johnny, Spike, Timmy
- Elicitation summary for the current expert (Johnny):
- Question ID: 1
- Question Text: How would you

Buttons: Cancel, OK

Figura 82: Finestra di elicitazione (1)

L'esperto può inserire le risposte e confermarle cliccando "OK" (Figura 83).

The window titled "Entity Elicitation: Obstacle4" shows the same questionnaire table as Figure 82, but with answers entered in the "Answer" column. A red rectangle highlights the "Answer" column.

Id	Text	Answer
1	How would you characterize the worst-case operating conditions related to this obstacle?	Answer 1...
2	How likely is this obstacle to occur in the worst-case operating conditions given its evidences?	0.6
3	How would you characterize the best-case operating conditions related to this obstacle?	Answer 2...
4	How likely is this obstacle to occur in the best-case operating conditions given its evidences?	0.8
5	How would you characterize the average-case operating conditions related to this obstacle?	Answer 3...

The "Entity Snippet" panel on the right displays the same information as in Figure 82.

Buttons: Cancel, OK

Figura 83: Finestra di elicitazione (2)

3.5.4.7 Salvataggio di una sessione

Cliccando su "Save" nella finestra principale, l'utente visualizza la finestra di salvataggio della sessione, in cui specifica la modalità con cui le modifiche devono essere rese persistenti (Figura 84).

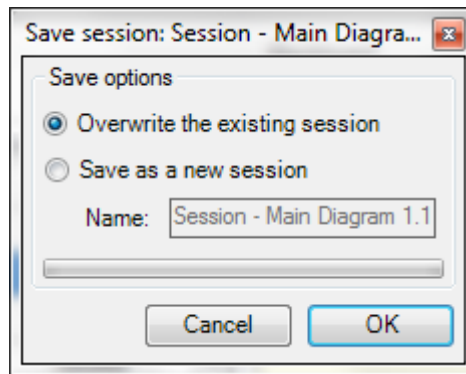


Figura 84: Finestra di salvataggio di una sessione

L'utente visualizza inoltre un warning ogni volta che apporta modifiche alla sessione, e, senza salvarla, effettua un'operazione che potrebbe causare inconsistenze nei dati come:

- Selezionare una nuova sessione
- Cambiare l'utente identificato nel sistema
- Esportare i dati della sessione
- Chiudere la finestra principale

Il sistema richiede pertanto all'utente se desidera rendere permanenti le modifiche effettuate alla sessione (Figura 85). In caso positivo, viene visualizzata la finestra di salvataggio della sessione.

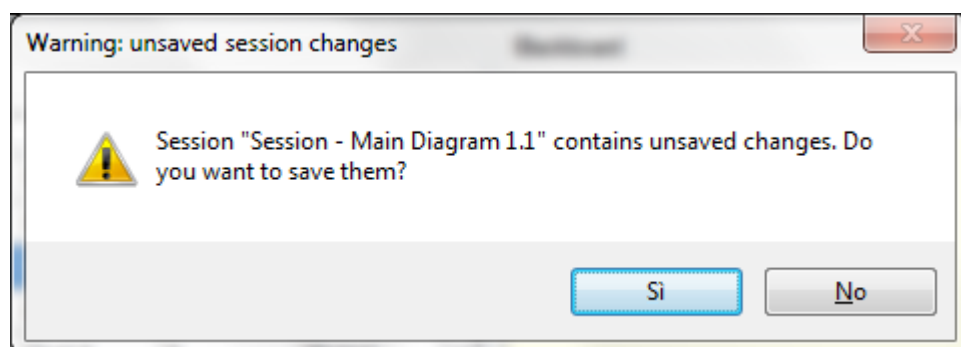


Figura 85: Avviso di modifiche non salvate

3.5.4.8 Export di una sessione

Cliccando su “Export” nella finestra principale, l’utente visualizza la finestra di export della sessione, in cui specifica le opzioni di export (Figura 86).

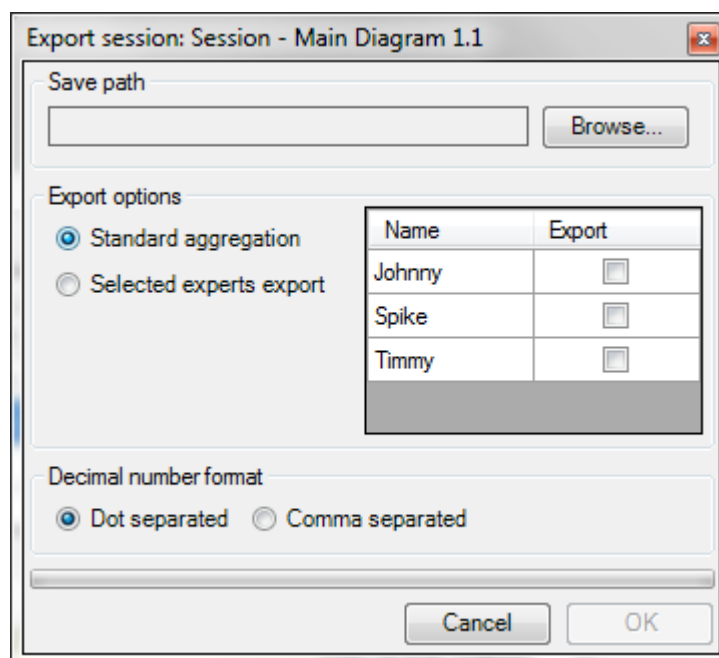


Figura 86: Finestra di export di una sessione (1)

Il bottone “OK” di conferma diventa attivo solo quando l’utente ha inserito correttamente tutti i parametri di export: path di salvataggio del file, esperti dei quali si desidera esportare i dati e formato di separazione dei numeri decimali (Figura 87).

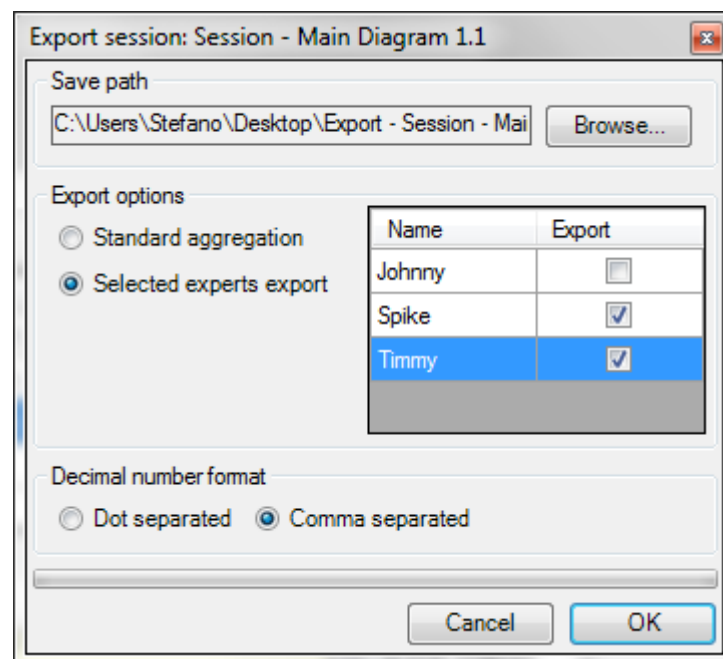


Figura 87: Finestra di export di una sessione (2)

4 Conclusioni e sviluppi futuri

FAV_QUOTE_PLACEHOLDER

Nel presente lavoro di Tesi Specialistica in Ingegneria Informatica si è affrontato il tema della qualificazione delle nuove tecnologie nell'ambito della gestione del rischio e della valutazione della sicurezza funzionale in sistemi tecnologici e safety-critical. La base teorica dello studio è stata la metodologia di qualificazione Modus [1] definita dal Simula Research Laboratory [2] e da DNV [3].

L'obiettivo principale dello studio consisteva nello sviluppo di uno strumento software che unificasse i vari aspetti di Modus in un'implementazione coerente. Lo strumento avrebbe dovuto fornire un ambiente di modellazione per la specifica degli obiettivi, delle interfacce per il supporto all'elicitazione delle opinioni degli esperti e alla loro memorizzazione persistente, nonché un supporto alla fase di analisi quantitativa delle probabilità oggetto di studio. Vincoli progettuali dello strumento consistevano nel limitarne l'esecuzione all'ambiente fornito da Enterprise Architect [5] e nella richiesta di interoperabilità con il tool @RISK [6].

Modus Tool, l'applicazione a supporto di Modus, è stata da noi sviluppata nel rispetto di modelli di processi software industriali come RUP [8], coprendo le fasi di ideazione, elaborazione e creazione. Particolare attenzione è stata posta alla gestione dei requisiti, dei casi d'uso ed alla specifica dell'architettura. Si è mantenuto, durante tutto il corso del lavoro, l'obiettivo di mantenere ottimi livelli di *usability* dello strumento, al fine di garantire un prodotto finale quanto più completo ed intuitivo possibile. Nello specifico, sono state realizzate le seguenti tre funzionalità principali.

- ***Supporto alla modellazione degli obiettivi (goal modeling)***: Modus Tool, efficacemente integrato in un ambiente MDE come Enterprise Architect, permette all'utente un'efficace ed intuitiva creazione di modelli di obiettivi strategici, permettendo la definizione degli stessi in più diagrammi complementari. La conformità dei diagrammi con la notazione KAOS [9], utilizzata da Modus, è stata assicurata dalla definizione di un profilo UML che consentisse di rappresentare tutte le astrazioni del dominio della modellazione per obiettivi all'interno di diagrammi UML standard.
- ***Validazione di modelli di obiettivi (model validation)***: Modus Tool consente all'utente di validare dal punto di vista strutturale un modello degli obiettivi, generando un report in cui sono riportati gli errori individuati ed i suggerimenti per la loro correzione. A causa della potenziale vastità di tali modelli, l'utente potrebbe non rispettare correttamente il formalismo di rappresentazione, generando modelli ambigui o inconsistenti che invaliderebbero le successive operazioni su di essi.
- ***Supporto all'elicitazione delle opinioni degli esperti (elicitation)***: Modus Tool fornisce un supporto considerevole durante la fase di elicitazione delle opinioni degli esperti, in termini di separazione delle responsabilità delle varie figure professionali coinvolte nell'attività, controllo della consistenza dei dati, comunicazione tra gli utenti, modalità di condivisione delle informazioni e presentazione delle stesse. A causa dell'iteratività del processo di elicitazione, si è definito un framework operativo basato su *sessioni* di lavoro, in modo da consentire il mantenimento di uno storico delle attività precedenti utile al monitoraggio di dati immessi in diversi periodi temporali e alla loro eventuale rivisitazione. Modus Tool consente inoltre di esportare i dati elicitati dagli esperti in un formato interpretabile da @RISK, al fine di consentire ivi il proseguimento del processo definito da Modus con la fase di analisi probabilistica.

4.1 Sviluppi futuri

Come accade in ogni metodologia innovativa e di recente ideazione, lo studio condotto in Modus lascia spazio a diverse direzioni che lavori successivi potrebbero intraprendere. In tal senso, è anche possibile definire eventuali estensioni future di Modus Tool fra cui quelle appresso elencate.

- ***Supporto per la modellazione e l'aggregazione di diverse dimensioni di decomposizione degli obiettivi:*** allo stato attuale, Modus non definisce delle linee guida per combinare i diversi tipi di decomposizione degli obiettivi (strutturale, comportamentale e di processo). Una volta condotto uno studio in questo senso, la funzionalità di goal modeling offerta da Modus Tool potrebbe supportare anche questa caratteristica, aumentando l'espressività e la chiarezza del modello degli obiettivi.
- ***Supporto per la mitigazione degli errori di elicitazione:*** Modus Tool offre già un supporto basilare all'importante attività di gestione di *bias* presenti nelle opinioni degli esperti. Vengono infatti mitigati:
 - Gli errori causati da opinioni errate che derivano da influenze reciproche che potrebbero avere gli esperti nel caso in cui siano a conoscenza delle risposte altrui: il tool consente la visualizzazione di dati globali riguardo un'elicitazione solo dopo esplicita conferma da parte dell'utente;
 - Gli errori causati dalla stanchezza degli esperti: il tool fornisce un meccanismo basato su sessioni di lavoro che è possibile interrompere e riprendere in momenti successivi;
 - Gli errori causati da eventuali stime approssimative delle quantità: il tool presenta agli esperti questionari di elicitazione in cui bisogna anche motivare le risposte fornite ad ogni domanda.

Ulteriori sviluppi in questa direzione potrebbero riguardare tecniche avanzate come l'analisi basata sulla correlazione dei dati forniti dagli

esperti, al fine di individuare possibili forzature in merito alla coerenza di opinioni.

- ***Supporto per la comparazione quantitativa di costi e performance tra componenti in alternativa nella fase di design:*** Modus è stato definito come una metodologia per la qualificazione di nuove tecnologie, piuttosto che come attività a supporto della progettazione delle stesse. L'approccio presentato potrebbe però essere utilizzato anche al fine di comparare, in fase di progettazione, il grado di soddisfacimento degli obiettivi che deriva dall'utilizzo di diverse componenti: Modus tool potrebbe in tal senso offrire un supporto alla modellazione di diverse scelte alternative al fine di determinare quale, fissata una soglia di costo, massimizzi gli obiettivi strategici.

Bibliografia

- [1] M. Sabetzadeh et al., *Modus: A Goal-Based Approach for Quantitative Assessment of Technical Systems*.
- [2] Simula Research Laboratory. Simula. [Online]. <http://simula.no/>
- [3] Det Norske Veritas. DNV. [Online]. <http://www.dnv.com/>
- [4] Simula. ModelME! Project. [Online]. <http://modelme.simula.no/>
- [5] Sparx Systems. Enterprise Architect. [Online]. <http://www.sparxsystems.com.au/>
- [6] Palisade. @RISK. [Online]. <http://www.palisade.com/risk/>
- [7] Microsoft. Microsoft Excel. [Online]. <http://office.microsoft.com/en-us/excel/>
- [8] P. Kruchten, *The Rational Unified Process: An Introduction*.
- [9] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*.
- [10] W. Schafer and H. Wehrheim, *The Challenges of Building Advanced Mechatronic Systems*.
- [11] P. Feiler et al., *Ultra-Large.Scale Systems: The Software Challenge of the Future*.
- [12] A. Avizienis, J.C. Laprie, and B. Randell, *Fundamental Concepts of Dependability*.
- [13] "Functional safety of electrical/electronic/programmable electronic safety-

related systems," IEC, IEC 61508,.

- [14] M. Swanson, "Security Self-Assessment Guide for Information Technology Systems,".
- [15] T. Kelly and R. Weaver, "The Goal Structuring Notation - A Safety Argument Notation,".
- [16] M. Sabetzadeh et al., *Combining Goal Models, Expert Elicitation, and Probabilistic Simulation for Qualification of New Technology*.
- [17] Det Norske Veritas, *Qualification Procedures for New Technology*.
- [18] D. Jackson et al., *Software for Dependable Systems: Sufficient Evidence?*.
- [19] C. Ericson, *Hazard Analysis Techniques for System Safety*.
- [20] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*.
- [21] E. Yu, *Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering*.
- [22] A. O'Hagan et al., *Uncertain Judgements: Eliciting Experts' Probabilities*.
- [23] M. Meyer and J. Booker, *Eliciting and Analyzing Expert Judgment: A Practical Guide*.
- [24] D. Vose, *Quantitative Risk Analysis: A Guide to Monte Carlo Simulation Modelling*.
- [25] M. Morgan and M. Henrion, *Uncertainty: A guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*.
- [26] C. Spetzler and C. Steal Von Holstein, *Exceptional Paper-Probability Encoding in*

Decision Analysis.

- [27] P. Garvey, *Probability Methods for Cost Uncertainty Analysis.*
- [28] B. Freimut, L. Briand, and F. Vollei, *Determining Inspection Cost-Effectiveness by Combining Project Data and Expert Opinion.*
- [29] T. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.*
- [30] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, *Formal Reasoning Techniques for Goal Models.*
- [31] A. Mosleh, *Interaction Between Model and Data in Common Cause Failure Analysis.*
- [32] C. Robert and G. Casella, *Monte Carlo Statistical Methods.*
- [33] J. Devore and N. Farnum, *Applied Statistics for Engineers and Scientists.*
- [34] OMG. Object Management Group. [Online]. <http://www.omg.org/>
- [35] OMG. SysML.org. [Online]. <http://www.sysml.org/>
- [36] OMG. MARTE. [Online]. <http://www.omgmarTE.org/>
- [37] W3C. XML. [Online]. <http://www.w3.org/XML/>
- [38] Microsoft. Visual Studio. [Online]. <http://www.microsoft.com/visualstudio/>

A Appendice A: Specifica dei requisiti utente di Modus Tool

FAV_QUOTE_PLACEHOLDER

A.1 Goal Modeling

Vengono riportati di seguito, reattivamente alla feature di Goal Modeling di Modus Tool, il diagramma dei requisiti (Figura 88) ed un estratto del documento dei requisiti utente (Tabella 8).

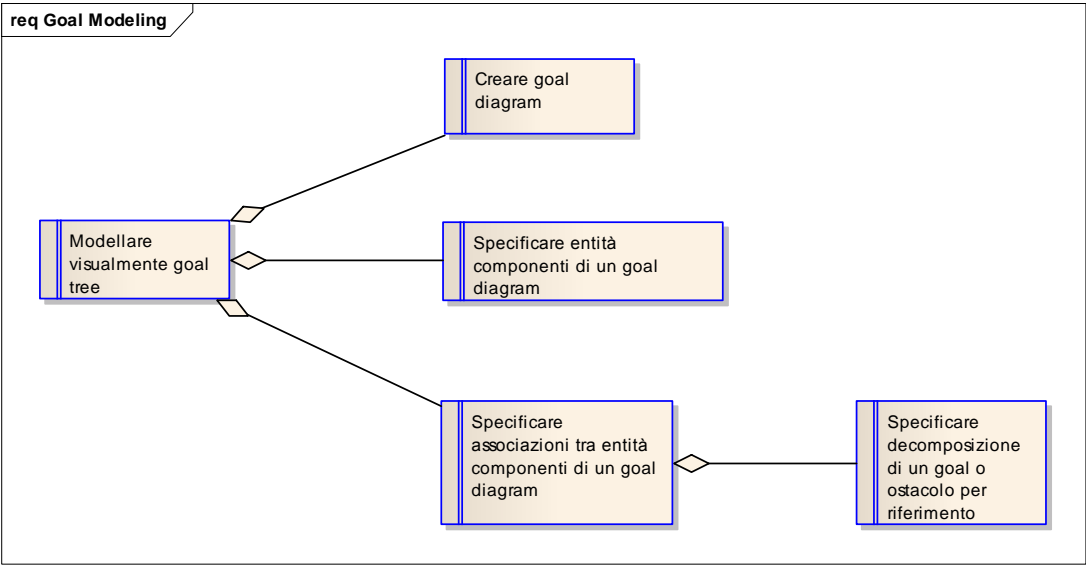


Figura 88: Diagramma dei requisiti della feature di Goal Modeling

1Modellare visualmente goal tree			
«Functional»	Status: Proposed	Priority: Medium	Difficulty: Medium

	<i>Phase: 1. 0</i>	<i>Version: 1. 0</i>	Il sistema deve offrire all'utente la possibilità di modellare visualmente goal tree, definendo le entità che lo compongono e le relazioni tra esse
1. 1 Creare goal diagram			
«Functional»	<i>Status: Proposed</i>	<i>Priority: Medium</i>	<i>Difficulty: Medium</i>
	<i>Phase: 1. 0</i>	<i>Version: 1. 0</i>	Il sistema deve offrire all'utente la possibilità di creare goal diagram che conterranno entità e relazioni tra esse
1. 2 Specificare associazioni tra entità componenti di un goal diagram			
«Functional»	<i>Status: Proposed</i>	<i>Priority: Medium</i>	<i>Difficulty: Medium</i>
	<i>Phase: 1. 0</i>	<i>Version: 1. 0</i>	Il sistema deve specificare le associazioni e le decomposizioni presenti tra le entità componenti di un goal diagram:
			<ul style="list-style-type: none"> • decomposizioni di goal/ostacoli attraverso ostruzioni ed AND/OR completi e parziali • relazioni tra decomposizioni ed assunzioni • relazioni tra goal/ostacoli ed evidenze • relazioni tra evidenze ed artefatti
«Functional»	<i>Status: Proposed</i>	<i>Priority: Medium</i>	<i>Difficulty: Medium</i>
	<i>Phase: 1. 0</i>	<i>Version: 1. 0</i>	Il sistema deve offrire all'utente la possibilità di specificare la decomposizione di un goal/ostacolo in un goal diagram diverso da quello in cui il goal/ostacolo è rappresentato
1. 3 Specificare entità componenti di un goal diagram			
«Functional»	<i>Status: Proposed</i>	<i>Priority: Medium</i>	<i>Difficulty: Medium</i>
	<i>Phase: 1. 0</i>	<i>Version: 1. 0</i>	Il sistema deve specificare le entità che compongono un goal diagram:
			<ul style="list-style-type: none"> • goal • ostacoli • assunzioni • evidenze • artefatti

Tabella 8: Requisiti utente della feature di Goal Modeling

A.2 Model Validation

Vengono riportati di seguito, relativamente alla feature di Model Validation di Modus Tool, il diagramma dei requisiti (Figura 89) ed un estratto del documento dei requisiti utente (Tabella 9).

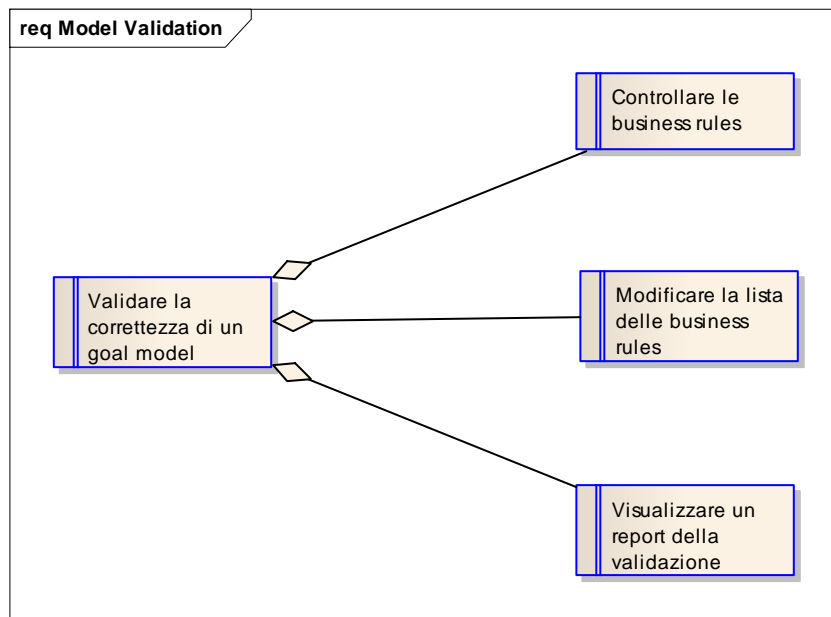


Figura 89: Diagramma dei requisiti della feature di Model Validation

1Validare la correttezza di un goal model

«Functional»

Status: Proposed

Priority: Medium

Difficulty: Medium

Phase: 1. 0

Version: 1. 0

Il sistema deve validare la correttezza di un goal model secondo delle regole preimpostate, presentando all'utente un report della validazione

1. 1Controllare le consistency rules

«Functional»

Status: Proposed

Priority: Medium

Difficulty: Medium

Phase: 1. 0

Version: 1. 0

Il sistema deve validare i goal diagram controllando un insieme di regole preimpostate, definito in un apposito file, che esprimono i vincoli che le entità componenti di un goal model devono rispettare

1. 2Modificare la lista delle consistency rules

«Functional»

Status: Proposed

Priority: Medium

Difficulty: Medium

Phase: 1. 0

Version: 1. 0

Il sistema deve offrire all'utente la possibilità di modificare la lista

delle regole di consistenza controllate ad ogni validazione, scegliendo quali regole devono essere verificate.

1.3 Visualizzare un report della validazione

«Functional»

Status: Proposed

Priority: Medium

Difficulty: Medium

Phase: 1.0

Version: 1.0

Il sistema deve visualizzare, dopo il completamento dell'operazione, un report della validazione in cui sono indicati:

- Le regole controllate durante la validazione
- L'esito del controllo di ogni regola
- Il tempo di validazione
- Gli elementi analizzati
- Gli errori individuati

Tabella 9: Requisiti utente della feature di Model Validation

Ai requisiti utente è inoltre necessario allegare un documento che specifichi le le consistency rules di validazione, che sono identificabili come *business rules* proprie del dominio oggetto di studio. L'effort complessivo relativo alla realizzazione della funzionalità della feature di Model Validation è stato in considerevole parte impiegato per la definizione dell'insieme di regole, in origine non definito esaurientemente, al pari del formalismo di goal modeling (Sezione C.1.3).

Dopo un'attenta analisi svolta in più iterazioni, si è stesa una lista di 34 regole di consistenza divise in 6 gruppi (Tabella 10).

Gruppo 1		Local (connectors) rules
Applicato a		Goal Diagram
Descrizione		The goal diagram must be syntactically correct
Regole	#	Descrizione
	1	Each AND branch link must connect either a goal or an obstacle to either a complete AND node or partial AND node
	2	Each AND anchor link must connect a complete AND node or partial AND node to either a goal or an obstacle
	3	Each complete OR link must connect either a goal to another goal, or an obstacle to another obstacle
	4	Each partial OR link must connect either a goal to another goal, or an

	obstacle to another obstacle																		
5	Each obstruction link must connect either a goal to an obstacle, or an obstacle to a goal, or an obstacle or goal to a complete AND node																		
6	Each assumption link must connect an assumption to a complete AND node																		
7	Each evidence link must either connect a goal or obstacle to an evidence																		
8	Each artifact link must connect an evidence to an artifact																		
9	Each reference link must connect a goal or obstacle to a UML Diagram hyperlink																		
Gruppo 2	Global (decomposition) rules																		
Applicato a	Goal Diagram																		
Descrizione	The goal diagram must be semantically correct																		
Regole	<table> <tr> <th>#</th><th>Descrizione</th></tr> <tr> <td>1</td><td>Each goal or obstacle which decomposes through a complete AND cannot decompose through any other links</td></tr> <tr> <td>2</td><td>Each goal or obstacle which decomposes through a partial AND cannot decompose through any other links</td></tr> <tr> <td>3</td><td>Each goal or obstacle which decomposes through a complete OR can only decompose through other complete OR links</td></tr> <tr> <td>4</td><td>Each goal or obstacle which decomposes through a partial OR can only decompose through other partial OR links</td></tr> <tr> <td>5</td><td>Each goal or obstacle which is obstructed by an obstacle or goal cannot decompose in any way other than that obstruction link</td></tr> <tr> <td>6</td><td>Each goal or obstacle which decomposes through a complete AND must decompose in at least two other entities</td></tr> <tr> <td>7</td><td>Each goal or obstacle which decomposes through a complete OR must decompose in at least two other entities</td></tr> <tr> <td>8</td><td>Each goal or obstacle which decomposes through a complete AND must decompose in at least a goal or obstacle</td></tr> </table>	#	Descrizione	1	Each goal or obstacle which decomposes through a complete AND cannot decompose through any other links	2	Each goal or obstacle which decomposes through a partial AND cannot decompose through any other links	3	Each goal or obstacle which decomposes through a complete OR can only decompose through other complete OR links	4	Each goal or obstacle which decomposes through a partial OR can only decompose through other partial OR links	5	Each goal or obstacle which is obstructed by an obstacle or goal cannot decompose in any way other than that obstruction link	6	Each goal or obstacle which decomposes through a complete AND must decompose in at least two other entities	7	Each goal or obstacle which decomposes through a complete OR must decompose in at least two other entities	8	Each goal or obstacle which decomposes through a complete AND must decompose in at least a goal or obstacle
#	Descrizione																		
1	Each goal or obstacle which decomposes through a complete AND cannot decompose through any other links																		
2	Each goal or obstacle which decomposes through a partial AND cannot decompose through any other links																		
3	Each goal or obstacle which decomposes through a complete OR can only decompose through other complete OR links																		
4	Each goal or obstacle which decomposes through a partial OR can only decompose through other partial OR links																		
5	Each goal or obstacle which is obstructed by an obstacle or goal cannot decompose in any way other than that obstruction link																		
6	Each goal or obstacle which decomposes through a complete AND must decompose in at least two other entities																		
7	Each goal or obstacle which decomposes through a complete OR must decompose in at least two other entities																		
8	Each goal or obstacle which decomposes through a complete AND must decompose in at least a goal or obstacle																		

	9	Each goal or obstacle which decomposes through a partial AND must decompose in at least a goal or obstacle
	10	Each complete or partial AND which is connected via an AND anchor link to a goal must have each of its AND branch links connected to a goal
	11	Each complete or partial AND which is connected via an AND anchor link to an obstacle must have each of its AND branch links connected to an obstacle
	12	Each complete or partial AND which is connected via an AND anchor link to a goal must have its obstruction link, if any, connected to an obstacle
	13	Each complete or partial AND which is connected via an AND anchor link to an obstacle must have its obstruction link, if any, connected to a goal
Gruppo 3	Tree-structure rules	
Applicato a	Goal Diagram	
Descrizione	The goal decomposition graph must be a tree with the root node being a goal, the non-leaf nodes being goals, obstacles or assumptions and the leaf nodes being evidences	
Regole	#	Descrizione
	1	There must be exactly one root entity of the tree
	2	The root entity of the tree must be a goal
	3	Each non-root entity of the tree must have exactly one father
	4	Each non-leaf entity of the tree must be either a goal, or an obstacle, or a complete AND, or a partial AND
	5	Each leaf entity of the tree must be an evidence
	6	The tree must have at least a leaf entity
Gruppo 4	Evidence-Artifact multiplicity rules	

Applicato a	Goal Diagram		
Descrizione	Artifacts and evidences must support each other		
Regole	#	Descrizione	
	1	Each evidence must be supported by exactly one artifact	
	2	Each artifact must support at least one evidence	
Gruppo 5	Reference correctness rules		
Applicato a	Goal Diagram		
Descrizione	Each decomposition referenced by a diagram hyperlink must be semantically correct w. r. t. the source decomposed element		
Regole	#	Descrizione	
	1	Each goal or obstacle connected via a reference link to a diagram hyperlink cannot decompose in any other way other than that reference link	
	2	Each goal or obstacle connected via a reference link to a diagram hyperlink must be the root element of the tree of the goal decomposition diagram linked by that diagram hyperlink	
Gruppo 6	Other rules		
Descrizione	N/A		
Regole	#	Applicata a	Descrizione
	1	Goal Model	Each element must appear in at least one diagram;
	2	Goal Diagram	Each diagram referenced by a diagram hyperlink must comply with all the rules

Tabella 10: Regole di consistenza

Come è possibile osservare, quasi tutte le regole vengono controllate su oggetti di tipo goal diagram. Solo l'ultimo gruppo, contenente regole di vario genere che non

è possibile catalogare altrimenti, presenta una regola controllata su oggetti di tipo goal model.

Nessuna regola viene controllata direttamente su oggetti di tipo entità, rendendo non necessaria un'eventuale funzionalità di controllo mirato di singoli elementi.

A.3 Elicitation

Vengono riportati di seguito, reattivamente alla feature di Elicitation di Modus Tool, il diagramma dei requisiti (Figura 90) ed un estratto del documento dei requisiti utente (Tabella 11).

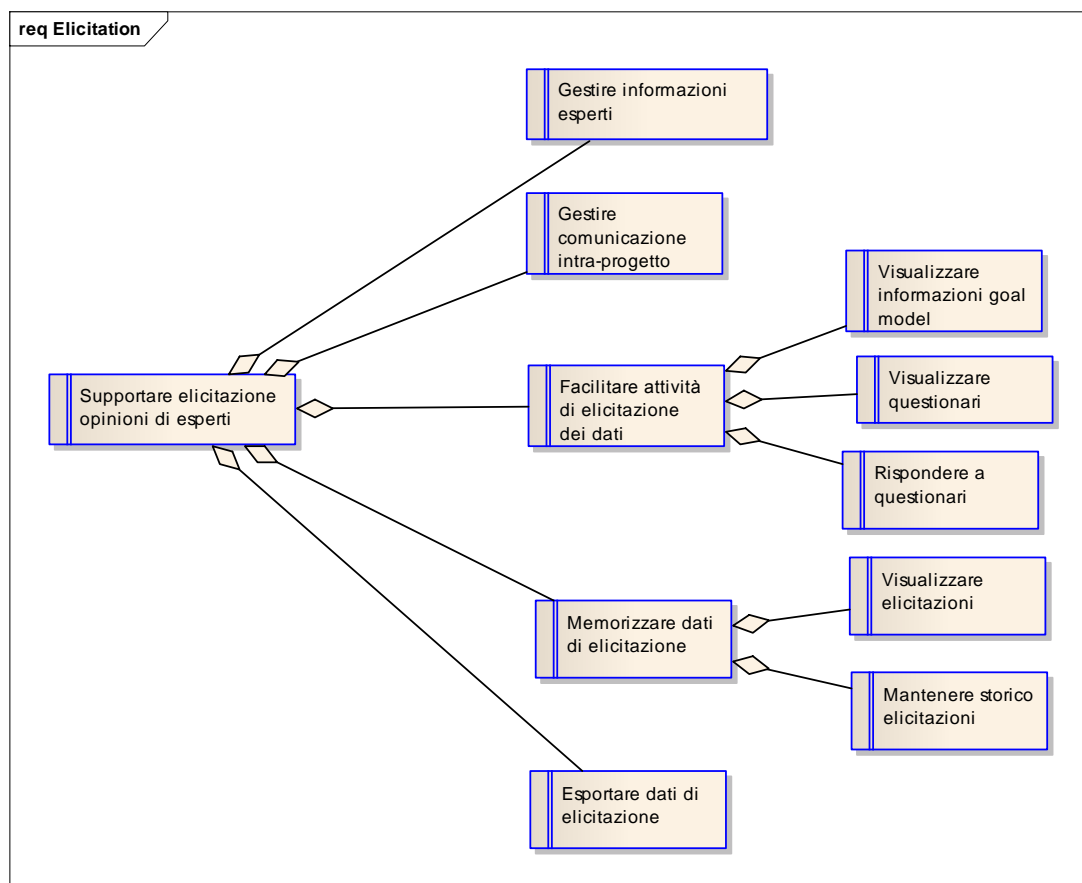


Figura 90: Diagramma dei requisiti della feature di Elicitation

1Supportare elicitazione opinioni di esperti

«Functional»

Status: Proposed *Priority:* Medium *Difficulty:* Medium
Phase: 1. 0 *Version:* 1. 0

Il sistema deve supportare l'attività di elicitazione delle opinioni degli esperti riguardo le distribuzioni di probabilità di goal ed ostacoli foglia nei goal diagram, consentendo all'utente di rispondere a questionari strutturati e revisionare le proprie risposte

1. 1Esportare dati di elicitazione

«Functional»

Status: Proposed *Priority:* Medium *Difficulty:* Medium
Phase: 1. 0 *Version:* 1. 0

Il sistema deve esportare i dati di elicitazione di un goal tree in un formato compatibile con il tool "@RISK" al fine di effettuare la propagazione delle distribuzioni, specificando gli esperti di cui si vuole considerare la probabilità ed aggregando i loro dati di elicitazione

1. 2Facilitare attività di elicitazione dei dati

«Functional»

Status: Proposed *Priority:* Medium *Difficulty:* Medium
Phase: 1. 0 *Version:* 1. 0

Il sistema deve facilitare l'attività di elicitazione dei dati, consentendo all'utente di rispondere a dei questionari strutturati riguardo la probabilità di eventi e di visualizzare le informazioni sul goal model di riferimento

1. 2. 1Rispondere a questionari

«Functional»

Status: Proposed *Priority:* Medium *Difficulty:* Medium
Phase: 1. 0 *Version:* 1. 0

Il sistema deve offrire all'utente la possibilità di rispondere a questionari strutturati di elicitazione, consentendo la modifica temporanea degli stessi ed il salvataggio permanente dei risultati

1. 2. 2Visualizzare informazioni goal model

«Functional»

Status: Proposed *Priority:* Medium *Difficulty:* Medium
Phase: 1. 0 *Version:* 1. 0

Il sistema deve visualizzare informazioni sul goal model contestualmente alla presentazione all'utente dei questionari di elicitazione:

- Entità in esame
- Numero di esperti che hanno esaminato l'entità
- Informazioni di decomposizione dell'entità
- Evidenze ed artefatti correlati all'entità (se elemento)
- Su esplicita richiesta dell'utente, informazioni sulle elicitazioni degli altri esperti sull'entità in esame

1. 2. 3Visualizzare questionari

«Functional» **Status:** Proposed **Priority:** Medium **Difficulty:** Medium
Phase: 1. 0 **Version:** 1. 0

Il sistema deve visualizzare dei questionari strutturati di elicitazione, memorizzando le precedenti risposte confermate dagli utenti

1. 3Gestire comunicazione intra-progetto

«Functional» **Status:** Proposed **Priority:** Medium **Difficulty:** Medium
Phase: 1. 0 **Version:** 1. 0

Il sistema deve offrire la possibilità agli utenti che partecipano all'attività di elicitazione di comunicare tra loro

1. 4Gestire informazioni esperti

«Functional» **Status:** Proposed **Priority:** Medium **Difficulty:** Medium
Phase: 1. 0 **Version:** 1. 0

Il sistema deve gestire le informazioni rilevanti riguardo gli esperti che partecipano ad un'attività di elicitazione, memorizzando dati personali come background, expertise e livello di educazione

1. 5Memorizzare dati di elicitazione

«Functional» **Status:** Proposed **Priority:** Medium **Difficulty:** Medium
Phase: 1. 0 **Version:** 1. 0

Il sistema deve memorizzare persistentemente i risultati dell'attività di elicitazione dei dati

1. 5. 1Mantenere storico elicitazioni

«Functional» **Status:** Proposed **Priority:** Medium **Difficulty:** Medium
Phase: 1. 0 **Version:** 1. 0

Il sistema deve mantenere uno storico delle attività passate di elicitazione, memorizzando le risposte precedentemente confermate e visualizzando lo status dei goal model di riferimento al momento dell'elicitazione

1. 5. 2Visualizzare elicitazioni

«Functional» **Status:** Proposed **Priority:** Medium **Difficulty:** Medium
Phase: 1. 0 **Version:** 1. 0

Il sistema deve visualizzare in dettaglio i dati di elicitazione di distribuzioni di probabilità di tutti gli utenti, specificando

- Goal diagram di riferimento
- Data di elicitazione
- Entità di riferimento dell'elicitazione
- Statistiche di elicitazione relative all'entità: risposte fornite e dettaglio degli esperti coinvolti

Tabella 11: Requisiti utente della feature di Elicitation

B Appendice B: Specifica dei casi d'uso di Modus Tool

B.1 Goal Modeling

Vengono riportati di seguito, relativamente alla feature di Goal Modeling di Modus Tool, il diagramma dei casi d'uso con la specifica delle feature realizzate (Figura 91) ed un estratto del documento di specifica dei casi d'uso (Tabella 12).

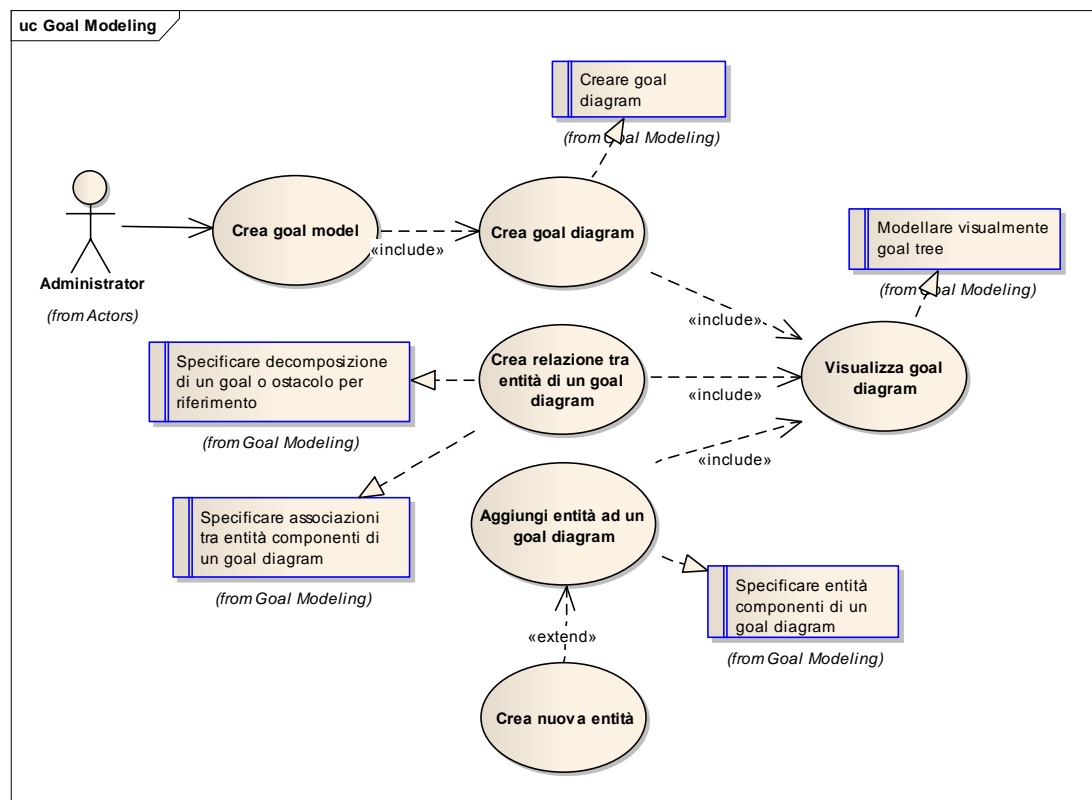


Figura 91: Use case diagram della feature di Goal Modeling

Use Case:	Aggiungi entità ad un goal diagram	Use Case Id:	
Status:	Proposed	Version:	1.0
		Phase:	1.0

Author:	Stefano	Created:	21/03/20	Modified:	26/03/2011
			11		
Notes:	Questo Use Case describe la possibilità offerta all'utente di aggiungere un'entità ad un goal diagram				
Related To:	Source	Connector	Target		
	Goal Modeling. Aggiungi entità ad un goal diagram	<i>Realization</i>	Goal Modeling. Specificare entità componenti di un goal diagram		
	Goal Modeling. Crea nuova entità	<i>Extend</i>	Goal Modeling. Aggiungi entità ad un goal diagram		
	Goal Modeling. Aggiungi entità ad un goal diagram	<i>Include</i>	Goal Modeling. Visualizza goal diagram		
Scenarios:					
<u>Basic Path</u>		Questo scenario describe la sequenza dei passi relativi all'aggiunta di una nuova entità ad un goal diagram			
Flusso base					
1a. Aggiungi entità esistente					
1. L'utente trascina un'entità dalla toolbox "Modus" nella finestra principale dove è visualizzato il goal diagram, <i>Uses:</i>					
Exception: 1a. Aggiungi entità esistente					
2. <extend> Crea nuova entità, <i>Uses:</i>					
3. Il sistema aggiorna il repository specificando che l'entità selezionata è contenuta nel diagramma specificato, <i>Uses:</i>					
4. <include> Visualizza goal diagram, <i>Uses:</i>					
<u>Alternate</u>		Questo scenario describe la sequenza dei passi relativi alla funzionalità di aggiunta di una entità esistente ad un goal diagram			
Aggiungi entità esistente					
1. L'utente trascina un'entità dal project browser nella finestra principale dove è					

visualizzato il goal diagram, *Uses:*

Constraints:	Name	Status	Notes
	Goal diagram visualizzato	Approved	È stato visualizzato il goal diagram di appartenenza dell'entità

Post-Conditions:	Name	Status	Notes
	Entità aggiunta al goal diagram	Approved	L'entità è stata aggiunta al goal diagram

--

Use Case:	Crea goal diagram		Use Case Id:
Status:	Proposed	Version: 1.0	Phase: 1.0
Author:	Stefano	Created: 21/03/2011	Modified: 26/03/2011

Notes: Questo Use Case descrive la possibilità offerta all'utente di creare un goal diagram all'interno di un goal model

Related To:	Source	Connector	Target
	Goal Modeling. Crea goal diagram	<i>Realization</i>	Goal Modeling. Creare goal diagram
	Goal Modeling. Crea goal model	<i>Include</i>	Goal Modeling. Crea goal diagram
	Goal Modeling. Crea goal diagram	<i>Include</i>	Goal Modeling. Visualizza goal diagram

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi alla creazione di un goal diagram all'interno di un goal model

1. L'utente specifica un goal model e seleziona "Add Diagram", *Uses:*
2. Il sistema visualizza la finestra "New Diagram" di creazione del diagramma, *Uses:*

3. L'utente specifica il tipo "Goal Diagram" ed inserisce il nome del diagramma, *Uses:*

4. Il sistema aggiunge il nuovo goal diagram al repository, *Uses:*

5. <include> Visualizza goal diagram, *Uses:*

Constraints:	Name	Status	Notes
	Goal model esistente	Approved	È stato creato il goal model di appartenenza del goal diagram

Post-Conditions:	Name	Status	Notes
	Goal diagram creato	Approved	Il goal diagram è stato creato nel repository

--

Use Case:	Crea goal model		Use Case Id:
Status:	Proposed	Version: 1. 0	Phase: 1. 0
Author:	Stefano	Created:	Modified:
		21/03/20	26/03/2011
	11		

Notes: Questo Use Case descrive la possibilità offerta all'utente di creare un goal model all'interno di un root model

Related To:	Source	Connector	Target
	Goal Modeling. Crea goal model	<i>Include</i>	Goal Modeling. Crea goal diagram
	Actors. Administrator	<i>Association</i>	Goal Modeling. Crea goal model

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi alla creazione di un goal model all'interno di un root model di Enterprise Architect

1. L'utente specifica un root model e seleziona "Add a new model", *Uses:*
2. Il sistema visualizza la finestra "Select model(s)" di creazione del modello, *Uses:*
3. L'utente specifica il tipo "Goal Model", *Uses:*
4. Il sistema crea il goal model e lo aggiunge al repository, *Uses:*
5. <include> Crea goal diagram, *Uses:*

Constraints:	Name	Status	Notes
	Root model esistente	Approved	È stato creato il root model di appartenenza del goal model

Post-Conditions:	Name	Status	Notes
	Goal model creato	Approved	Il goal model è stato creato nel repository

--

Use Case:	Crea nuova entità		Use Case Id:
Status:	Proposed	Version: 1. 0	Phase: 1. 0
Author:	Stefano	Created: 21/03/2011	Modified: 26/03/2011
	11		

Notes: Questo Use Case descrive la possibilità offerta all'utente di creare un'entità nel repository di Enterprise Architect

Related To:	Source	Connector	Target
	Goal Modeling. Crea nuova entità	<i>Extend</i>	Goal Modeling. Aggiungi entità ad un goal diagram

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi alla creazione di una nuova entità

1. L'utente specifica un goal model e seleziona "Add Element", *Uses:*

2. Il sistema visualizza la finestra "New Element" di creazione dell'entità, [Uses:](#)
3. L'utente specifica il tipo "Class", lo stereotipo relativo, ed inserisce il nome dell'entità, [Uses:](#)
4. Il sistema aggiunge la nuova entità al repository, e la visualizza nel project browser, [Uses:](#)

Constraints:	Name	Status	Notes
	Goal model creato	Approved	È stato creato il goal model di appartenenza dell'entità

Post-Conditions:	Name	Status	Notes
	Entità creata	Approved	L'entità è stata creata nel repository

Use Case:	Crea relazione tra entità di un goal diagram	Use Case Id:
------------------	--	--------------

Status:	Proposed	Version: 1. 0	Phase: 1. 0
----------------	----------	---------------	-------------

Author:	Stefano	Created:	Modified:
		21/03/20	26/03/2011

11

Notes:	Questo Use Case descrive la possibilità offerta all'utente di creare una relazione tra due entità di un goal diagram
---------------	--

Related To:	Source	Connector	Target
	Goal Modeling. Crea relazione tra entità di un goal diagram	Realization	Goal Modeling. Specificare decomposizione di un goal o ostacolo per riferimento
	Goal Modeling. Crea relazione tra entità di un goal diagram	Realization	Goal Modeling. Specificare associazioni tra entità componenti di un goal diagram
	Goal Modeling. Crea relazione tra entità di un goal diagram	Include	Goal Modeling. Visualizza goal diagram

Scenarios:Basic Path**Flusso base**

Questo scenario descrive la sequenza dei passi relativi alla creazione di una nuova relazione tra entità esistenti all'interno di un goal diagram

1. L'utente trascina una relazione dalla toolbox "Modus" nella finestra principale dove è visualizzato il goal diagram, specificando l'entità sorgente e l'entità destinazione della relazione, *Uses:*
2. Il sistema aggiorna il repository specificando che l'entità selezionata è contenuta nel diagramma specificato, *Uses:*
3. <include> Visualizza goal diagram, *Uses:*

Constraints:	Name	Status	Notes
	Goal diagram visualizzato	Approved	È stato visualizzato il goal diagram di appartenenza della relazione
	Entità esistenti	Approved	Sono state create ed aggiunte al goal diagram le entità che partecipano alla relazione

Post-Conditions:	Name	Status	Notes
	Relazione creata	Approved	La relazione tra le entità è stata creata nel repository

Use Case:	Visualizza goal diagram		Use Case Id:
Status:	Proposed	Version: 1. 0	Phase: 1. 0
Author:	Stefano	Created: 21/03/20	Modified: 26/03/2011
11			

Notes: Questo Use Case descrive la possibilità offerta all'utente di visualizzare un goal diagram nella finestra principale di Enterprise Architect

Related To:	Source	Connector	Target
--------------------	---------------	------------------	---------------

	Goal Modeling. Aggiungi entità ad un goal diagram	<i>Include</i>	Goal Modeling. Visualizza goal diagram
	Goal Modeling. Crea goal diagram	<i>Include</i>	Goal Modeling. Visualizza goal diagram
	Goal Modeling. Crea relazione tra entità di un goal diagram	<i>Include</i>	Goal Modeling. Visualizza goal diagram
	Goal Modeling. Visualizza goal diagram	<i>Realization</i>	Goal Modeling. Modellare visualmente goal tree
Scenarios:			
<u><i>Basic Path</i></u>		Questo scenario descrive la sequenza dei passi relativi alla visualizzazione di un goal diagram selezionato nel project browser di Enterprise Architect	
Flusso base			
1a. Naviga decomposizione per riferimento			
1. L'utente seleziona un goal diagram nel project browser, <i>Uses:</i>		<i>Exception:</i> 1a. Naviga decomposizione per riferimento	
2. Il sistema visualizza il goal diagram selezionato nella finestra principale, <i>Uses:</i>			
<u><i>Alternate</i></u>		Questo scenario descrive la sequenza dei passi relativi alla funzionalità di visualizzazione di un goal diagram di cui è stato selezionato l'hyperlink all'interno di un altro goal diagram	
Naviga decomposizione per riferimento			
1. L'utente seleziona un hyperlink di un diagramma che rappresenta una decomposizione per riferimento all'interno del diagramma visualizzato, <i>Uses:</i>			
Constraints:	Name	Status	Notes
	Goal diagram esistente	Approved	È stato creato il goal diagram da visualizzare
Post-Conditions:	Name	Status	Notes

Goal diagram visualizzato	Approved	Il goal diagram è visualizzato nella finestra principale
---------------------------	----------	--

Tabella 12: Use case report della feature di Goal Modeling

B.2 Model Validation

Vengono riportati di seguito, relativamente alla feature di Model Validation di Modus Tool, il diagramma dei casi d'uso con la specifica delle feature realizzate (Figura 92) ed un estratto del documento di specifica dei casi d'uso (Tabella 13).

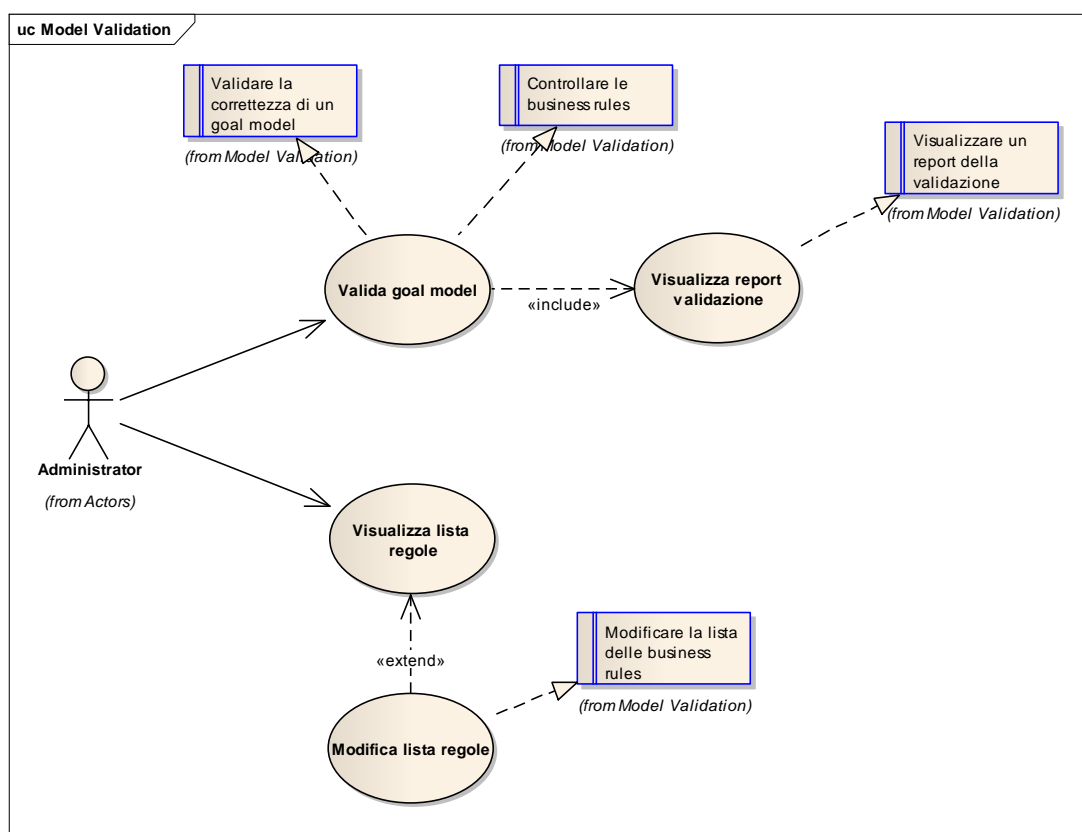


Figura 92: Use case diagram della feature di Model Validation

Use Case:	Modifica lista regole	Use Case Id:	
Status:	Proposed	Version:	1. 0
Author:	Stefano	Created:	Modified:

Notes: Questo Use Case descrive la possibilità offerta all'utente di modificare la lista delle regole controllate durante la validazione di un goal model

Related To:	Source	Connector	Target
	Model Validation. Modifica lista regole	<i>Dependency</i>	Model Validation. Visualizza lista regole
	Model Validation. Modifica lista regole	<i>Realization</i>	Model Validation. Modificare la lista delle business rules
	Actors. Administrator	<i>Association</i>	Model Validation. Modifica lista regole
	Model Validation. Modifica lista regole	<i>Include</i>	Model Validation. Visualizza lista regole

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi alla modifica del file contenente le regole controllate durante la validazione di un goal model

1. <include> Visualizza lista regole, *Uses:*
2. L'utente modifica il file contenente le regole, e seleziona "Salva", *Uses:*
3. Il sistema salva le modifiche effettuate dall'utente sul disco rigido, *Uses:*

Constraints:	Name	Status	Notes
	File delle regole esistente	Approved	Il file con la lista delle regole è presente nella memoria della macchina dove è in esecuzione il sistema

Use Case:	Valida goal model		Use Case Id:	
Status:	Proposed	Version:	1. 0	Phase: 1. 0
Author:	Stefano	Created:	25/03/20	Modified: 26/03/2011
		11		
Notes:	Questo Use Case descrive la possibilità offerta all'utente di validare un goal model secondo un insieme di regole preimpostate			
Related To:	Source	Connector	Target	
	Model Validation. Valida goal model	<i>Realization</i>	Model Validation. Controllare le business rules	
	Model Validation. Valida goal model	<i>Realization</i>	Model Validation. Validare la correttezza di un goal model	
	Actors. Administrator	<i>Association</i>	Model Validation. Valida goal model	
	Model Validation. Valida goal model	<i>Include</i>	Model Validation. Visualizza report validazione	
Scenarios:				
<u>Basic Path</u>	Questo scenario descrive la sequenza dei passi relativi alla validazione di un goal model			
Flusso base				
1. L'utente specifica un goal model nel project browser e seleziona "Validate selected", <i>Uses:</i>				
2. Il sistema carica il file contenente le regole da controllare ed avvia la validazione del goal model, <i>Uses:</i>				
3. Il sistema stampa nella finestra di output il report della validazione, <i>Uses:</i>				
4. <include> Visualizza report validazione, <i>Uses:</i>				
Constraints:	Name	Status	Notes	
	Goal model esistente	Approved	Il goal model è presente nel	

File delle regole Approved esistente			repository Il file delle regole è presente nella memoria della macchina su cui è in esecuzione il sistema
Use Case:	Visualizza lista regole		Use Case Id:
Status:	Proposed	Version: 1. 0	Phase: 1. 0
Author:	Stefano	Created: 25/03/2011	Modified: 26/03/2011
11			
Notes:	Questo Use Case descrive la possibilità offerta all'utente di visualizzare la lista delle regole controllate durante la validazione di un goal model		
Related To:	Source	Connector	Target
	Model Validation. Modifica lista regole	Dependency	Model Validation. Visualizza lista regole
	Model Validation. Modifica lista regole	Include	Model Validation. Visualizza lista regole
Scenarios:			
Basic Path Flusso base		Questo scenario descrive la sequenza dei passi relativi alla visualizzazione del file contenente le regole controllate durante la validazione di un goal model	
1. L'utente naviga nel disco rigido fino alla cartella che contiene il file delle regole e lo apre, Uses:			
2. Il sistema visualizza nell'editor di default il contenuto del file delle regole, Uses:			
Constraints:	Name	Status	Notes
	File delle regole Approved esistente		Il file con la lista delle regole è presente nella memoria della macchina dove è in

Post-Conditions:	Name	Status	Notes
	File delle regole visualizzato	Approved	Il file delle regole è visualizzato nell'editor di default del sistema

Use Case:	Visualizza report validazione	Use Case Id:	
Status:	Proposed	Version:	1. 0
Author:	Stefano	Created:	25/03/2011
		Modified:	26/03/2011
	11		

Notes: Questo Use Case descrive la possibilità offerta all'utente di visualizzare un report relativo alla validazione di un modello

Related To:	Source	Connector	Target
	Model Validation. Visualizza report validazione	<i>Realization</i>	Model Validation. Visualizzare un report della validazione
	Model Validation. Valida goal model	<i>Include</i>	Model Validation. Visualizza report validazione

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi alla visualizzazione del report della validazione di un goal model

1. L'utente seleziona la finestra "Output" di Enterprise Architect, *Uses:*
2. Il sistema visualizza il report della validazione del goal model, riportando gli errori individuati, gli eventuali suggerimenti per la correzione degli stessi e le statistiche di validazione, *Uses:*

Constraints:	Name	Status	Notes
	Validazione	Approved	La validazione del goal model

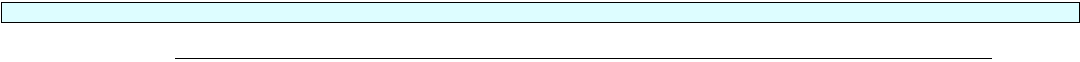
terminata	è terminata
	

Tabella 13: Use case report della feature di Model Validation

B.3 Elicitation

Vengono riportati di seguito, relativamente alla feature di Model Validation di Modus Tool, il diagramma dei casi d'uso con la specifica delle feature realizzate (Figura 93) ed un estratto del documento di specifica dei casi d'uso (Tabella 14).

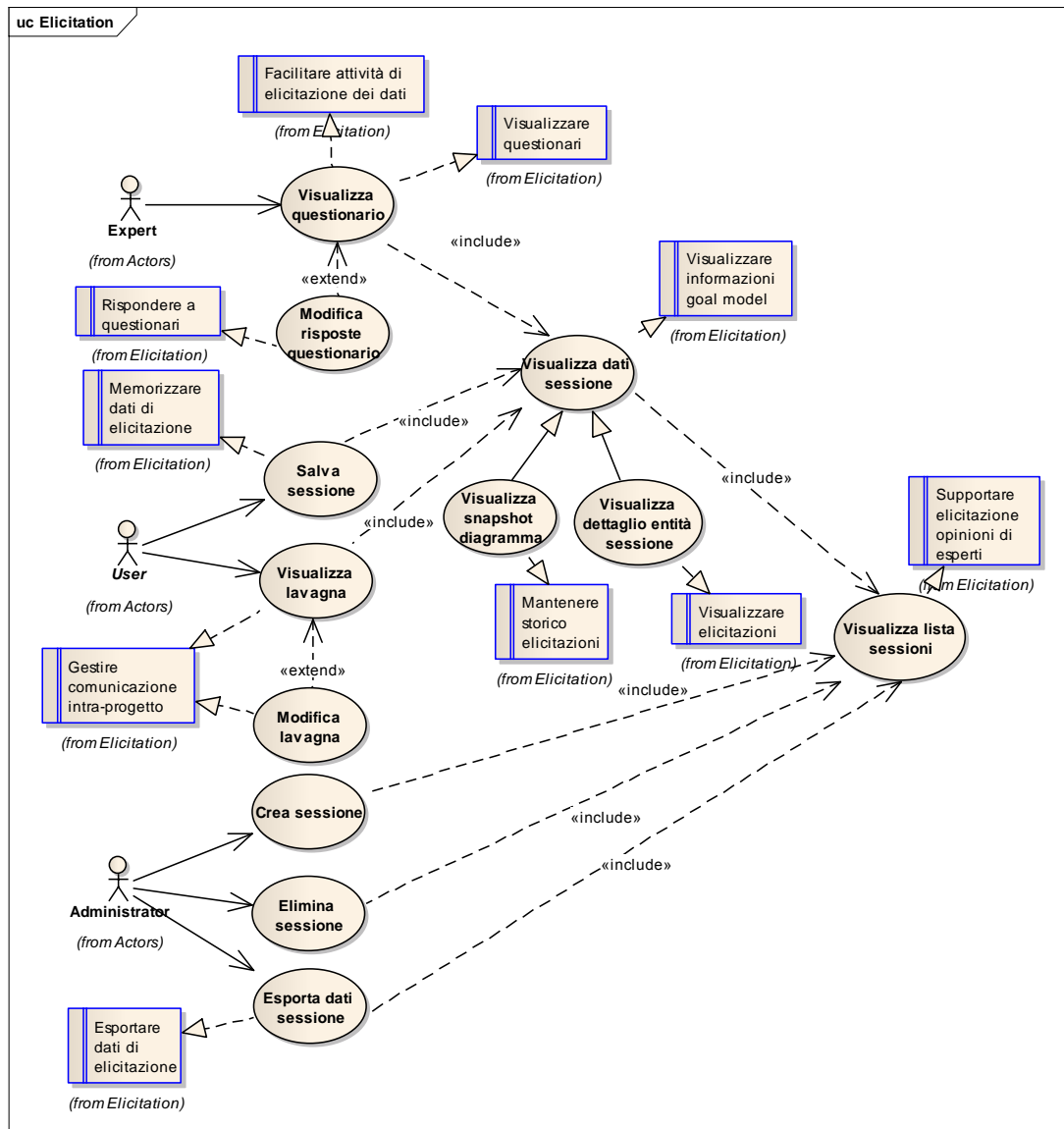


Figura 93: Use case diagram della feature di Elicitation

Use Case:	Crea sessione		Use Case Id:
Status:	Proposed	Version:	1.0
Author:	Stefano	Created:	27/03/2011
			27/03/2011
			011
Notes:	Questo Use Case descrive la possibilità offerta all'utente di creare una nuova sessione di elicitazione		
Related To:	Source	Connector	Target
	Elicitation. Crea sessione	Include	Elicitation. Visualizza lista sessioni

Actors. Administrator		Association	Elicitation. Crea sessione						
Precondition:	<table><tr><th>Name</th><th>Status</th><th>Notes</th></tr><tr><td>Goal diagram validato</td><td>Approved</td><td>Il goal diagram relativo alla sessione da creare è stato validato con successo</td></tr></table>	Name	Status	Notes	Goal diagram validato	Approved	Il goal diagram relativo alla sessione da creare è stato validato con successo		
Name	Status	Notes							
Goal diagram validato	Approved	Il goal diagram relativo alla sessione da creare è stato validato con successo							
Scenarios:									
<u>Basic Path</u>		Questo scenario descrive la sequenza dei passi relativi alla creazione di una nuova sessione							
Basic Path									
1. <include> Visualizza lista sessioni, Uses:									
2. L'utente seleziona "New session", Uses:									
3. Il sistema visualizza la finestra di creazione di una sessione, Uses:									
4. L'utente specifica un goal diagram nel project browser, inserisce il nome ed il tipo della sessione e seleziona "Ok", Uses:									
5. Il sistema crea una nuova sessione con i parametri specificati e chiude la finestra di creazione della sessione, Uses:									
6. <include> Visualizza lista sessioni, Uses:									
Post-Conditions:	<table><tr><th>Name</th><th>Status</th><th>Notes</th></tr><tr><td>Sessione creata</td><td>Approved</td><td>La sessione è presente nel repository</td></tr></table>	Name	Status	Notes	Sessione creata	Approved	La sessione è presente nel repository		
Name	Status	Notes							
Sessione creata	Approved	La sessione è presente nel repository							
Use Case:	Elimina sessione		Use Case Id:						
Status:	Proposed	Version: 1.0	Phase: 1.0						
Author:	Stefano	Created:	Modified: 27/03/2011						
27/03/2									
011									

Notes:	Questo Use Case descrive la possibilità offerta all'utente di eliminare una sessione di elicitazione esistente			
Related To:	Source	Connector	Target	
	Actors. Administrator	Association	Elicitation. sessione	Elimina
	Elicitation. sessione	Elimina Include	Elicitation. lista sessioni	Visualizza
Precondition:	Name	Status	Notes	
	Sessione esistente	Approved	La sessione è presente nel repository	
Scenarios:				
Basic Path		Questo scenario descrive la sequenza dei passi relativi all'eliminazione di una sessione esistente		
Flusso base				
1. <include> Visualizza lista sessioni, Uses:				
2. L'utente seleziona una sessione dalla lista e seleziona "Delete session", Uses:				
3. Il sistema visualizza una finestra di conferma dell'operazione, Uses:				
4. L'utente conferma l'operazione, Uses:				
5. Il sistema elimina la sessione selezionata, Uses:				
6. <include> Visualizza lista sessioni, Uses:				
Post-Conditions:	Name	Status	Notes	
	Sessione eliminata	Approved	La sessione non è presente nel repository	
Use Case:	Esporta dati sessione			Use Case Id:
Status:	Proposed	Version:	1. 0	Phase: 1. 0

Author:	Stefano	Created:	27/03/2011
			27/03/2
	011		
Notes:	Questo Use Case descrive la possibilità offerta all'utente di esportare i dati di una sessione di elicitazione esistente in formato XLS compatibile con @RISK		
Related To:	Source	Connector	Target
	Elicitation. Esporta dati sessione	Include	Elicitation. Visualizza lista sessioni
	Elicitation. Esporta dati sessione	Realization	Elicitation. Esportare dati di elicitazione
	Actors. Administrator	Association	Elicitation. Esporta dati sessione
Scenarios:			
<u>Basic Path</u>			
Flusso base		Questo scenario descrive la sequenza dei passi relativi all'export dei dati di una sessione nel caso in cui l'utente abbia specificato di voler aggregare le distribuzioni di probabilità fornite da tutti gli esperti che hanno preso parte alla sessione	
4a. Specifica dati da esportare			
1. <include> Visualizza lista sessioni, Uses:			
2. L'utente seleziona una sessione dalla lista e seleziona "Export session", Uses:			
3. Il sistema visualizza la finestra di export della sessione, Uses:			
4. L'utente specifica il nome ed il path di salvataggio del file di export, il formato dei numeri decimali, l'aggregazione standard delle distribuzioni di probabilità di tutti gli esperti che hanno partecipato alla sessione e seleziona "Ok", Uses:			
Exception: 4a. Specifica dati da esportare			
5. Il sistema crea il file secondo le impostazioni specificate e chiude la finestra di export della sessione, Uses:			
<u>Alternate</u>			
Specifica dati da esportare		Questo scenario descrive la sequenza dei passi relativi all'export dei dati di una	

sessione nel caso in cui l'utente abbia scelto gli esperti le cui distribuzioni di probabilità saranno aggregate ed esportate nel file

1. L'utente specifica il nome ed il path di salvataggio del file di export, il formato dei numeri decimali, l'aggregazione mirata delle distribuzioni di probabilità, seleziona dalla lista di tutti gli esperti che hanno partecipato alla sessione quelli di cui vuole esportare i dati e seleziona "Ok", *Uses:*

Constraints:	Name	Status	Notes
	Permessi in scrittura	Approved	Il sistema ha i permessi necessari per accedere alla memoria in scrittura

--

Use Case:	Modifica lavagna		Use Case Id:	
Status:	Proposed	Version:	1. 0	Phase: 1. 0
Author:	Stefano	Created:	27/03/2011	
			27/03/2011	
			011	

Notes: Questo Use Case descrive la possibilità offerta all'utente di modificare il contenuto della lavagna della sessione selezionata

Related To:	Source	Connector	Target
	Elicitation. Modifica lavagna	<i>Extend</i>	Elicitation. Visualizza lavagna
	Elicitation. Modifica lavagna	<i>Realization</i>	Elicitation. Gestire comunicazione intra-progetto

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi alla modifica della lavagna della sessione selezionata

1. L'utente seleziona l'area di testo della lavagna e ne modifica il contenuto, *Uses:*

2. Il sistema visualizza in tempo reale le modifiche apportate all'area di testo, *Uses:*

Constraints:	Name	Status	Notes
	Lavagna visualizzata	Approved	La lavagna è visualizzata nella finestra principale

Post-Conditions:	Name	Status	Notes
	Modifiche visualizzate	Approved	Le modifiche apportate dall'utente sono visualizzate nell'area di testo della lavagna
	Sessione marcata come modificata	Approved	La sessione viene marcata con lo status "modificato", che indica che esistono modifiche non salvate

--

--

Use Case:	Modifica risposte questionario			Use Case Id:	
Status:	Proposed	Version:	1. 0	Phase:	1. 0
Author:	Stefano	Created:	27/03/2011		
			27/03/2011		
			011		

Notes:	Questo Use Case descrive la possibilità offerta all'utente di modificare le proprie risposte delle domande di un questionario di elicitazione
--------	---

Related To:	Source	Connector	Target
	Elicitation. Modifica risposte questionario	<i>Extend</i>	Elicitation. Visualizza questionario
	Elicitation. Modifica risposte questionario	<i>Realization</i>	Elicitation. Rispondere a questionari

Scenarios:

[Basic Path](#)

Flusso base

1. L'utente seleziona una o più domande del questionario inserendo

correttamente le risposte relative e seleziona "Ok", *Uses:*

2. Il sistema memorizza temporaneamente le risposte inserite fino a che non sarà selezionata una nuova sessione di elicitazione, *Uses:*

Constraints:	Name	Status	Notes
	Questionario visualizzato	Approved	Il questionario è visualizzato nella finestra del questionario
	Sessione marcata come modificata	Approved	La sessione viene marcata con lo status "modificato", che indica che esistono modifiche non salvate

Post-Conditions:	Name	Status	Notes
	Modifiche visualizzate	Approved	Le modifiche apportate dall'utente sono visualizzate nell'area del dettaglio dei dati relativi all'elicitazione dell'elemento selezionato

--

Use Case:	Salva sessione			Use Case Id:	
Status:	Proposed	Version:	1. 0	Phase:	1. 0
Author:	Stefano	Created:	27/03/2011		
			27/03/2011		
			011		

Notes: Questo Use Case descrive la possibilità offerta all'utente di salvare permanentemente tutte le modifiche apportate ai dati di una sessione, secondo la modalità di sovrascrittura, o la modalità di salvataggio con backup

Related To:	Source	Connector	Target
	Elicitation. sessione	Salva <i>Include</i>	Elicitation. Visualizza dati sessione
	Actors. User	<i>Association</i>	Elicitation. Salva sessione
	Elicitation. sessione	Salva <i>Realization</i>	Elicitation. Memorizzare dati di elicitazione

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi al salvataggio di una sessione mantenendo una copia di backup

4a. Salva sovrascrivendo la sessione

1. <include> Visualizza dati sessione, *Uses:*
2. L'utente seleziona "Save session", *Uses:*
3. Il sistema visualizza la finestra di salvataggio della sessione, *Uses:*
4. L'utente seleziona l'opzione "Save with backup", inserisce il nome con cui desidera salvare la sessione e seleziona "Ok", *Uses:*
Exception: 4a. Salva sovrascrivendo la sessione
5. Il sistema salva una nuova nessione con i dati correntemente visualizzati, senza apportare modifiche all'originale, *Uses:*
6. <include> Visualizza dati sessione, *Uses:*

Alternate

Salva sovrascrivendo la sessione

Questo scenario descrive la sequenza dei passi relativi al salvataggio di una sessione sovrascrivendo quella esistente

1. L'utente seleziona l'opzione "Save overwriting" e seleziona "Ok", *Uses:*
2. Il sistema salva la sessione apportando tutte le modifiche, *Uses:*
3. <include> Visualizza dati sessione, *Uses:*

Use Case:	Visualizza dati sessione			Use Case Id:	
Status:	Proposed	Version:	1. 0	Phase:	1. 0
Author:	Stefano	Created:	27/03/2011	Modified:	27/03/2011
			011		

Notes: Questo Use Case describe la possibilità offerta all'utente di visualizzare il dettaglio dei dati di una sessione

Related To:	Source	Connector	Target
	Elicitation. Visualizza dettaglio sessione	<i>Generalization</i>	Elicitation. Visualizza dati sessione
	Elicitation. Visualizza questionario	<i>Include</i>	Elicitation. Visualizza dati sessione
	Elicitation. Salva sessione	<i>Include</i>	Elicitation. Visualizza dati sessione
	Elicitation. Visualizza dati sessione	<i>Realization</i>	Elicitation. Visualizzare informazioni goal model
	Elicitation. Visualizza dati sessione	<i>Include</i>	Elicitation. Visualizza lista sessioni
	Elicitation. Visualizza lavagna	<i>Include</i>	Elicitation. Visualizza dati sessione
	Elicitation. Visualizza snapshot diagramma	<i>Generalization</i>	Elicitation. Visualizza dati sessione

Scenarios:

Basic Path

Flusso base

Questo scenario describe la sequenza dei passi relativi alla visualizzazione dei dati di una sessione selezionata dall'utente nel caso in cui nessuna sessione sia stata precedentemente selezionata, o la sezione precedentemente selezionata non sia stata modificata dall'ultimo salvataggio

3a. Salvataggio implicito sessione

1. <include> Visualizza lista sessioni, *Uses:*
2. L'utente seleziona una sessione nella lista, *Uses:*
3. Il sistema visualizza tutte le informazioni relative alla sessione nella finestra principale, *Uses:*

Exception: 3a. Salvataggio implicito sessione

Alternate

Salvataggio implicito sessione

Questo scenario descrive la sequenza dei passi relativi alla visualizzazione dei dati di una sessione selezionata dall'utente nel caso in cui la sezione precedentemente selezionata sia stata modificata dall'ultimo salvataggio

1. Il sistema visualizza una finestra di warning avvisando l'utente che la sessione precedentemente selezionata è stata modificata ma non salvata, *Uses:*
2. L'utente sceglie se salvare le modifiche o annullarle, eventualmente richiamando il caso d'uso "Salva sessione", *Uses:*

Post-Conditions:	Name	Status	Notes
	Dati transienti eliminati	Approved	I dati transienti relativi ad eventuali sessioni precedentemente visualizzate e non salvate sono eliminati

Use Case:	Visualizza dettaglio entità sessione			Use Case Id:
Status:	Proposed	Version:	1. 0	Phase: 1. 0
Author:	Stefano	Created:	27/03/2011	
			011	

Notes: Questo Use Case descrive la possibilità offerta all'utente di visualizzare il dettaglio relativo ad un'entità del goal diagram di una sessione

Related To:	Source	Connector	Target
	Elicitation. Visualizza dettaglio sessione	<i>Generalization</i>	Elicitation. Visualizza dati sessione
	Elicitation. Visualizza dettaglio	<i>Realization</i>	Elicitation. Visualizzare elicitazioni

sessione

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi alla visualizzazione del dettaglio relativo alle informazioni di una entità di un goal diagram di una sessione nel caso in cui sia specificato di non visualizzare eventuali dati inseriti da altri utenti

1a. Visualizzazione dati altri esperti

1. L'utente seleziona un'entità dalla lista delle entità elicitabili del diagramma della sessione, *Uses:*

Exception: 1a. Visualizzazione dati altri esperti

2. Il sistema visualizza nella sezione "Entity snippet" le informazioni relative all'entità selezionata, incluso eventualmente gli ultimi dati inseriti dall'utente in merito all'entità, *Uses:*

Alternate

Visualizzazione dati altri esperti

Questo scenario descrive la sequenza dei passi relativi alla visualizzazione del dettaglio relativo alle informazioni di una entità di un goal diagram di una sessione nel caso in cui sia specificato di visualizzare eventuali dati inseriti da altri utenti

1. L'utente seleziona un'entità dalla lista delle entità elicitabili del diagramma della sessione e seleziona "Show other experts data", *Uses:*

2. Il sistema visualizza nella sezione "Entity snippet" le informazioni relative all'entità selezionata, incluso eventualmente gli ultimi dati inseriti dall'utente ed i dati inseriti da altri utenti in merito all'entità, *Uses:*

Constraints:	Name	Status	Notes
	Dati visualizzati	sessione Approved	I dati della sessione sono visualizzati nella finestra principale

Use Case:	Visualizza lavagna			Use Case Id:	
Status:	Proposed	Version:	1. 0	Phase:	1. 0
Author:	Stefano	Created:	27/03/2011	Modified:	27/03/2011
			011		
Notes:	Questo Use Case descrive la possibilità offerta all'utente di visualizzare il contenuto della lavagna della sessione selezionata				
Related To:	Source	Connector	Target		
	Elicitation. Modifica lavagna	<i>Extend</i>	Elicitation.	Visualizza lavagna	
	Elicitation. Visualizza lavagna	<i>Include</i>	Elicitation.	Visualizza dati sessione	
	Elicitation. Visualizza lavagna	<i>Realization</i>	Elicitation.	Gestire comunicazione intra-progetto	
	Actors. User	<i>Association</i>	Elicitation.	Visualizza lavagna	
Scenarios:					
<u>Basic Path</u>	Questo scenario descrive la sequenza dei passi relativi alla visualizzazione della lavagna della sessione selezionata				
Flusso base					
1. <include> Visualizza dati sessione, <i>Uses:</i>					
2. Il sistema visualizza in forma testuale la lavagna della sessione selezionata nella sezione "Blackboard", <i>Uses:</i>					
3. <extend> Modifica lavagna, <i>Uses:</i>					
Use Case:	Visualizza lista sessioni			Use Case Id:	

Status:	Proposed	Version:	1. 0	Phase:	1. 0
Author:	Stefano	Created:	27/03/2	Modified:	27/03/2011
011					
Notes:	Questo Use Case descrive la possibilità offerta all'utente di visualizzare la lista delle sessioni di un progetto di Enterprise Architect				
Related To:	Source	Connector	Target		
	Elicitation. sessione	Crea	Include	Elicitation. Visualizza lista sessioni	
	Elicitation. dati sessione	Esporta	Include	Elicitation. Visualizza lista sessioni	
	Elicitation. dati sessione	Visualizza	Include	Elicitation. Visualizza lista sessioni	
	Elicitation. lista sessioni	Visualizza	Realization	Elicitation. Supportare elicitazione opinioni di esperti	
	Elicitation. sessione	Elimina	Include	Elicitation. Visualizza lista sessioni	
Scenarios:					
<u>Basic Path</u>		Questo scenario descrive la sequenza dei passi relativi alla visualizzazione della lista delle sessioni nel caso in cui l'utente non sia identificato all'interno del sistema			
Flusso base		1a. Utente già identificato			
1. L'utente si identifica nel sistema scegliendo il proprio nome dalla lista degli utenti, Uses:		Exception: 1a. Utente già identificato			
2. Il sistema carica la lista delle sessioni memorizzate e la visualizza all'utente esponendo il dettaglio delle informazioni relative alla partecipazione dell'utente nella sessione, Uses:					
3. Il sistema visualizza la lista delle sessioni nella finestra principale, Uses:					

Alternate

Utente già identificato

Questo scenario descrive la sequenza dei passi relativi alla visualizzazione della lista delle sessioni nel caso in cui l'utente sia già identificato all'interno del sistema

1. L'utente seleziona la finestra principale, *Uses:*

Constraints:	Name	Status	Notes
	Utente registrato al sistema	Approved	L'utente è registrato al sistema
	Dati di Modus Tool integri	Approved	I dati di esecuzione del tool posti nel repository di Enterprise Architect non sono stati modificati dall'utente

Post-Conditions:	Name	Status	Notes
	Utente identificato nel sistema	Approved	

Use Case:	Visualizza questionario			Use Case Id:	
Status:	Proposed	Version:	1. 0	Phase:	1. 0
Author:	Stefano	Created:	27/03/2011		
			27/03/2011		
			011		

Notes: Questo Use Case descrive la possibilità offerta all'utente di visualizzare un questionario contenente domande e risposte relative all'elicitazione di un'opinione relativa ad un'entità del goal diagram di una sessione

Related To:	Source	Connector	Target
	Actors. Expert	<i>Association</i>	Elicitation. Visualizza questionario
	Elicitation. Visualizza questionario	<i>Include</i>	Elicitation. Visualizza dati sessione
	Elicitation. Modifica risposte questionario	<i>Extend</i>	Elicitation. Visualizza questionario

Elicitation. questionario	Visualizza	Realization	Elicitation. questionari
Elicitation. questionario	Visualizza	Realization	Elicitation. Facilitare attività di elicitazione dei dati

Scenarios:

Basic Path

Flusso base

Questo scenario descrive la sequenza dei passi relativi alla visualizzazione delle domande e delle risposte di un questionario di elicitazione relativo ad un'entità del goal diagram di una sessione

1. <include> Visualizza dati sessione, *Uses:*
2. L'utente seleziona un'entità dalla lista delle entità elicitabili del diagramma della sessione e seleziona "Elicit probabilities", *Uses:*
3. Il sistema visualizza la finestra contenente le domande del questionario relativo all'entità selezionata, corredato dalle ultime risposte fornite dall'utente, *Uses:*
4. <extend> Modifica risposte questionario, *Uses:*

Use Case:	Visualizza snapshot diagramma	Use Case Id:	
Status:	Proposed	Version:	1.0
Author:	Stefano	Created:	27/03/2011
			27/03/2011
			011

Notes: Questo Use Case descrive la possibilità offerta all'utente di visualizzare una snapshot di un diagramma di una sessione di elicitazione al momento dell'ultimo salvataggio della stessa

Related To:	Source	Connector	Target
	Elicitation. Visualizza snapshot diagramma	Realization	Elicitation. Mantenere storico elicitazioni

Elicitation. Visualizza snapshot diagramma		Generalization	Elicitation. Visualizza dati sessione
Scenarios:			
Basic Path		Questo scenario descrive la sequenza dei passi relativi alla visualizzazione di una thumbnail dell'immagine di un diagramma relativo ad una sessione di elicitazione, creata al momento dell'ultimo salvataggio della sessione	
Flusso base		1a. Visualizzazione zoom snapshot	
1. L'utente seleziona un diagramma dalla lista dei diagrammi della sessione, Uses:		<i>Exception:</i> 1a. Visualizzazione zoom snapshot	
2. Il sistema visualizza nel pannello "Snapshot" una thumbnail dell'immagine del diagramma salvata al momento dell'ultimo salvataggio della sessione, Uses:			
Alternate		Questo scenario descrive la sequenza dei passi relativi alla visualizzazione di uno zoom dell'immagine di un diagramma relativo ad una sessione di elicitazione, creata al momento dell'ultimo salvataggio della sessione	
Visualizzazione zoom snapshot			
1. L'utente seleziona un diagramma dalla lista dei diagrammi della sessione e seleziona "Zoom in", Uses:			
2. Il sistema visualizza la finestra di zoom dell'immagine del diagramma salvata al momento dell'ultimo salvataggio della sessione, Uses:			
Constraints:	Name	Status	Notes
	Dati sessione visualizzati	Approved	I dati della sessione sono visualizzati nella finestra principale

Tabella 14: Use case report della feature di Elicitation

C Appendice C: Specifica dell'architettura di Modus Tool

C.1 Goal Modeling

C.1.1 Modus Goal Model Template

Il template relativo alla creazione di un goal model all'interno di EA è stato definito esportando in XMI la definizione del tipo package creata con l'ausilio del wizard di EA.

```
<?xml version="1. 0" encoding="windows-1252"?>

<xmi: XMI xmi: version="2. 1" xmlns: uml="http: //schema. omg. org/spec/UML/2. 1"
xmlns: xmi="http: //schema. omg. org/spec/XMI/2. 1">

    <xmi: Documentation exporter="Enterprise Architect"
exporterVersion="6. 5"/>

    <uml: Model xmi: type="uml: Model" name="EA_Model"
visibility="public">

        <packagedElement xmi: type="uml: Package" xmi:
id="EAPK_E3F54885_65F0_4b3a_AB03_1D5198527952" name="Goal Model"
visibility="public"/>

    </uml: Model>

    <xmi: Extension extender="Enterprise Architect" extenderID="6. 5">

        <elements>

            <element xmi:
idref="EAPK_E3F54885_65F0_4b3a_AB03_1D5198527952" xmi: type="uml: Package"
name="Goal Model" scope="public">

                <model
package2="EAID_E3F54885_65F0_4b3a_AB03_1D5198527952"
package="EAPK_4C9C5B07_9C97_491c_AD93_3100350256D2" tpos="11" ea_localid="16"
ea_eleType="package"/>

                <properties
isSpecification="false" sType="Package" nType="0" scope="public"/>

                <project phase="1. 0"
created="2010-11-18 15: 38: 48" modified="2010-11-18 15: 39: 24" complexity="1"
```

```

status="Proposed"/>

                                <code/>

                                <style
appearance="BackColor=-1; BorderColor=-1; BorderWidth=-1; FontColor=-1;
VSwimLanes=1; HSwimLanes=1; BorderStyle=0; "/>

                                <modelDocument/>

                                <tags/>

                                <xrefs/>

                                <extendedProperties

tagged="0" package_name="Model"/>

                                <packageproperties

tpos="11"/>

                                <paths/>

                                <times created="2010-11-
18 15: 38: 48" modified="2010-11-18 15: 39: 24" lastloaddate="2010-11-18 15: 38: 48"
lastsavedate="2010-11-18 15: 38: 48"/>

                                <flags
iscontrolled="FALSE" isprotected="FALSE" batchsave="0" batchload="0" usedtd="FALSE"
logxml="FALSE" packageFlags="isModel=1; VICON=0; "/>

                                </element>

                                </elements>

                                <connectors/>

                                <primitivetypes>

                                <packagedElement xmi: type="uml:
Package" xmi: id="EAPrimitiveTypesPackage" name="EA_PrimitiveTypes_Package"
visibility="public"/>

                                </primitivetypes>

                                <profiles/>

                                <diagrams>

                                <diagram xmi:
id="EAID_103FCA06_98C2_4c63_82D2_0BB11FD77C78">

                                <model
package="EAPK_E3F54885_65F0_4b3a_AB03_1D5198527952" localID="15"
owner="EAPK_E3F54885_65F0_4b3a_AB03_1D5198527952"/>

                                <properties name="Goal
Model" type="Logical"/>

                                <project author="Stefano
Di Alesio" version="1. 0" created="2010-11-18 15: 40: 10" modified="2010-11-18 15:
40: 10"/>

                                <style1
value="ShowPrivate=1; ShowProtected=1; ShowPublic=1; HideRelationships=0; Locked=0;
Border=1; HighlightForeign=1; PackageContents=1; SequenceNotes=0; ScalePrintImage=0;
PPgs. cx=0; PPgs. cy=0; DocSize. cx=826; DocSize. cy=1169; ShowDetails=0;
Orientation=P; Zoom=100; ShowTags=0; OpParams=1; VisibleAttributeDetail=0;
ShowOpRetType=1; ShowIcons=1; CollabNums=0; HideProps=0; ShowReqs=0; ShowCons=0;
PaperSize=9; HideParents=0; UseAlias=0; HideAtts=0; HideOps=0; HideStereo=0;
HideElemStereo=0; ShowTests=0; ShowMaint=0; ConnectorNotation=UML 2. 1;
ExplicitNavigability=0; AdvancedElementProps=1; AdvancedFeatureProps=1;

```

```

AdvancedConnectorProps=1; ShowNotes=0; SuppressBrackets=0; SuppConnectorLabels=0;
PrintPageHeadFoot=0; ShowAsList=0; "/>

<style2
value="MDGDgm=Goal Diagram: : GoalModeling; "/>

<swimlanes
value="locked=false; orientation=0; width=0; inbar=false; names=false; color=0;
bold=false; fcol=0; ; cls=0; "/>

<matrixitems
value="locked=false; matrixactive=false; swimlanesactive=true; width=1; "/>

<extendedProperties/>

</diagram>

</diagrams>

</xmi: Extension>

</xmi: XMI>

```

Tabella 15: Definizione del template di Goal Modeling

Non si ritiene questa la sede per una disamina approfondita dei relativi dettagli tecnologici, che comunque esulano dagli obiettivi accademici della presente tesi. Ci si limita ad evidenziare nel codice riportato in Tabella 15 alcune definizioni chiave. Nell'ordine:

- Dichiarazione del tipo di package "Goal Model"
- Definizione del tipo di package "Goal Model" come package contenente "Logical Diagram", cioè generici diagrammi di analisi del comportamento di un sistema
- Definizione del tipo di diagramma "Goal Diagram" come diagramma di default relativo al tipo di package "Goal Model"

C.1.2 Modus Goal Diagram Profile

Come accennato nella Sezione C.1.2, il tipo di diagramma "Goal Diagram" si configura come "Logical Diagram". La metaclassa "Diagram Logical" (Figura 94) è stata infatti decorata con gli attributi relativi al tipo di diagramma definito:

- Alias, che ne specifica la nomenclatura
- DiagramID, che ne specifica l'acronimo

- FrameString, che ne specifica il nome visualizzato nella barra di stato di EA
- Toolbox, che ne specifica il nome della toolbox aperta automaticamente ogni volta che un diagramma viene visualizzato nella finestra principale

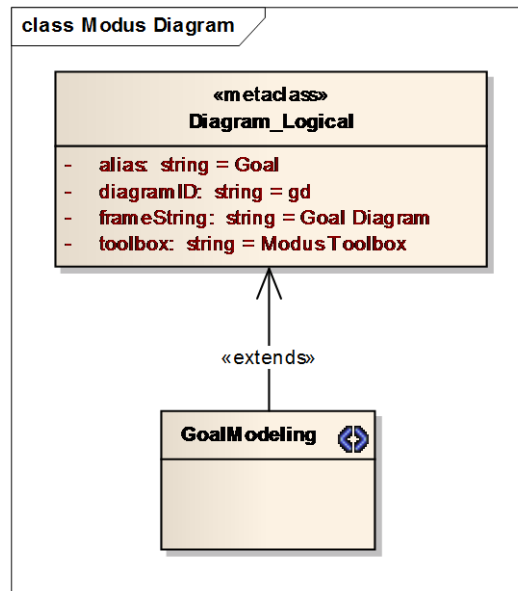


Figura 94: Profilo di definizione del tipo di diagramma “Goal Diagram”

C.1.3 Modus Entities Profile

Le entità utilizzate nelle feature di Goal Modeling ed Elicitation sono state definite come stereotipi UML che estendono le metaclassi “Class” e “Association”. Le Figura 95, Figura 96, Figura 97, Figura 98 e Figura 99 e la Tabella 16 riportano un riepilogo degli stereotipi definiti dal profilo delle entità, corredati dalla metaclassa estesa.

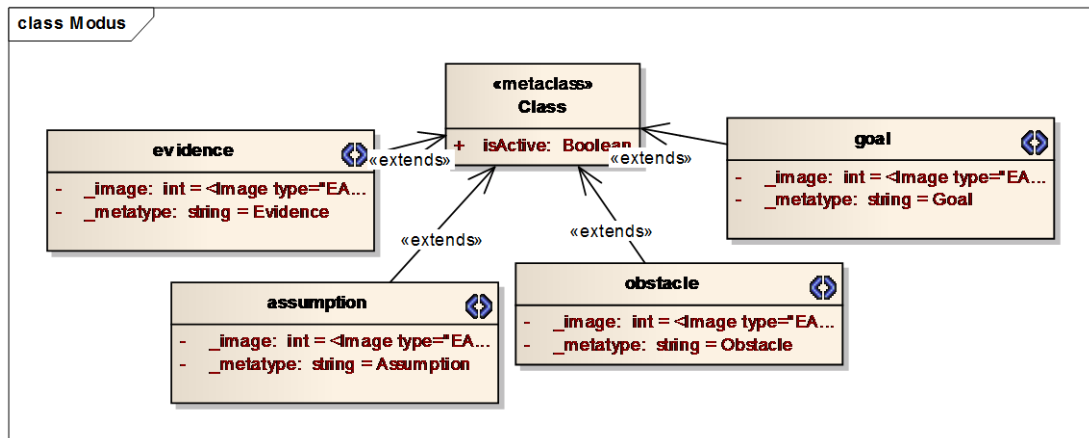


Figura 95: Profilo di definizione delle entità di Modus Tool (stereotipi di Class)

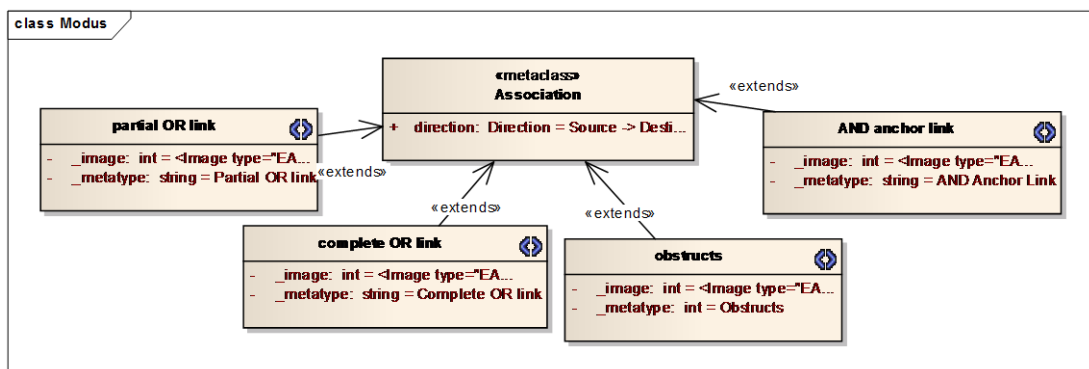


Figura 96: Profilo di definizione delle entità di Modus Tool (stereotipi di Association con l'attributo directed)

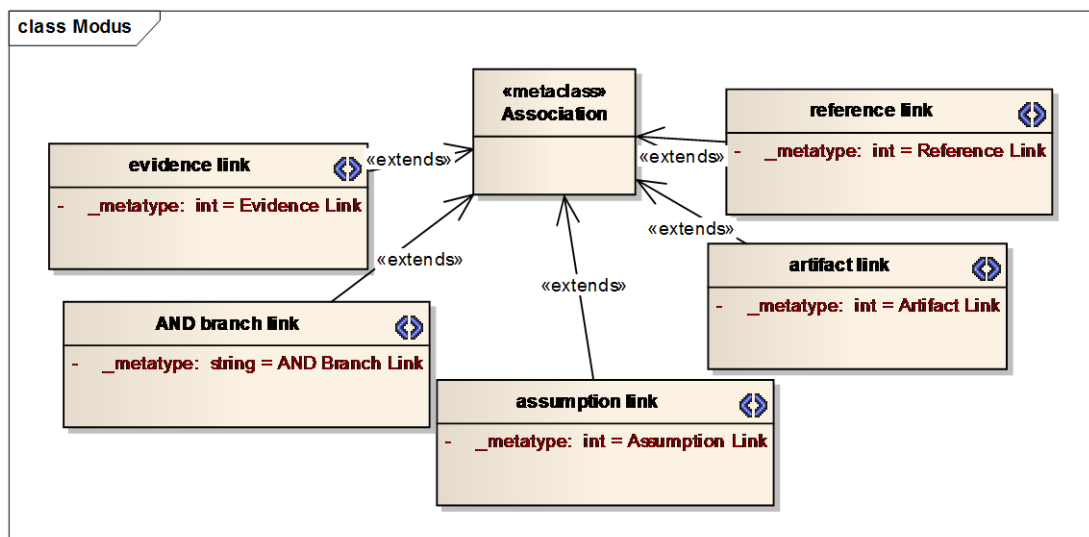


Figura 97: Profilo di definizione delle entità di Modus Tool (stereotipi di Association)

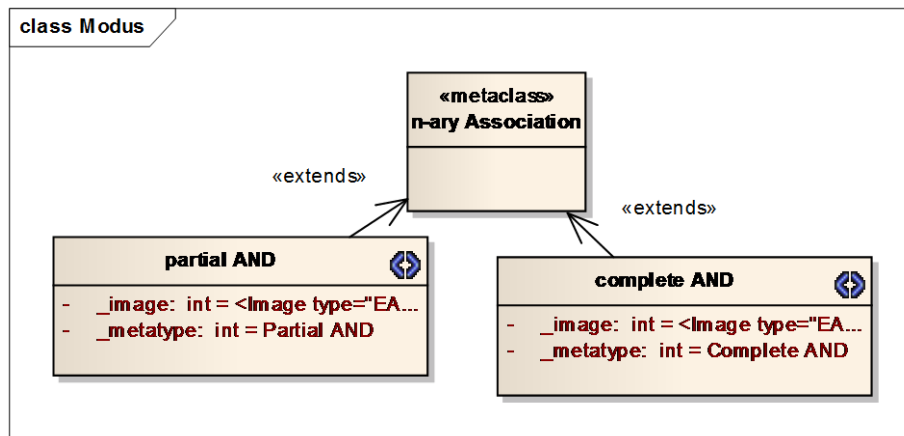


Figura 98: Profilo di definizione delle entità di Modus Tool (stereotipi di n-ary Association)

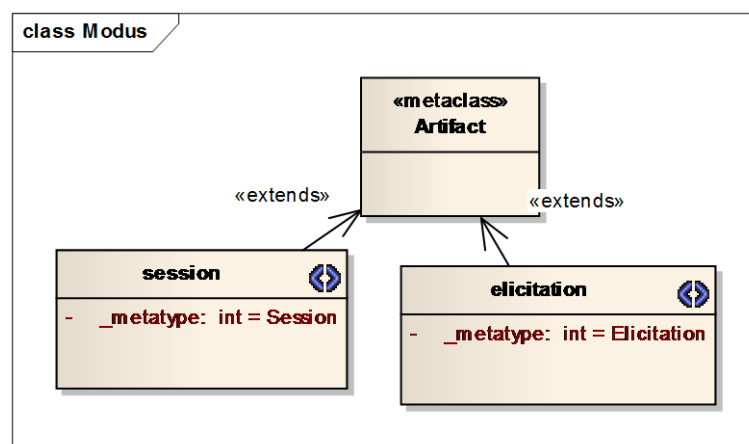


Figura 99: Profilo di definizione delle entità di Modus Tool (stereotipi di Artifact)

Stereotipo	Metaclassa	Utilizzo
«goal»	Class	Goal Modeling
«obstacle»	Class	Goal Modeling
«evidence»	Class	Goal Modeling
«assumption»	Class	Goal Modeling
«artifact»	N/A	Goal Modeling
«complete AND»	n-ary Association	Goal Modeling
«partial AND»	n-ary Association	Goal Modeling
«complete OR»	Association (directed)	Goal Modeling

«partial OR»	Association (directed)	Goal Modeling
«obstructs»	Association (directed)	Goal Modeling
«AND anchor link»	Association (directed)	Goal Modeling
«AND branch link»	Association (undirected)	Goal Modeling
«assumption link»	Association (undirected)	Goal Modeling
«evidence link»	Association (undirected)	Goal Modeling
«artifact link»	Association (undirected)	Goal Modeling
«reference link»	Association (undirected)	Goal Modeling
«session»	Artifact	Elicitation
«elicitation»	Artifact	Elicitation

Tabella 16: Tabella riepilogativa del profilo UML delle entità di Modus Tool

Agli stereotipi propri di elementi relativi alla feature di Goal Modeling, che sono pertanto visualizzati all'interno di diagrammi, sono stati associati degli Shape Script (Sezione C.1.4) il cui riferimento è riportato nell'attributo “_image”. Per gli stereotipi delle relazioni di decomposizione, la metaclassa “Association” è stata decorata con l'attributo “direction” che ne specifica la navigabilità.

È necessario precisare che il profilo definito non ha la pretesa di essere definito formalmente, né di rappresentare effettivamente un'estensione semantica di UML riutilizzabile *off the shelf*, come accade invece per profili come SysML [35] e MARTE [36]. Piuttosto, il *Modus Profile* è un compromesso tra un approccio standard che consente un buon grado di interoperabilità con Enterprise Architect, ed una soluzione ad-hoc che in grado di soddisfare i requisiti utente della feature di Goal Modeling. I termini entro cui il profilo delle entità si configura come soluzione ad-hoc, piuttosto che puramente formale, sono riportati di seguito.

Con l'esclusione degli stereotipi «session» ed «elicitation», riportati per completezza ma discussi nella Sezione C.3.1.4, nessuno stereotipo definisce dei *tagged values*, configurandosi come un semplice *marker* relativo ad una classe. L'esistenza del profilo è comunque giustificata dalla necessità di presentare delle forme *domain specific* di rappresentazione degli elementi sul diagramma.

Le relazioni di “complete AND” e “partial AND” sono state definite attraverso tre diversi stereotipi: un’associazione n-aria che ne qualifica il tipo, e due associazioni semplici stereotipate «AND anchor link» e «AND branch link». Tuttavia, la semantica di una decomposizione attraverso AND è identica a quella di una decomposizione attraverso OR, cioè quella di un’associazione semplice: è un solo goal (ostacolo) a scomporsi in uno o più goal (ostacoli). Questo accorgimento è stato necessario per far sì che la rappresentazione grafica delle decomposizioni attraverso AND fosse conforme a quella definita da Modus (Sezione 2.3.1).

Lo stereotipo «artifact», seppur non definito (non estende alcuna metaclassa), è comunque logicamente presente nel profilo ed è pertanto riportato per motivi di completezza espositiva. Nella pratica, gli artefatti vengono trascinati dal project browser o dall’ambiente di esecuzione di EA all’interno di un diagramma: ciò implica che essi non siano mai creati direttamente dall’utente all’interno di Enterprise Architect.

C.1.4 Shape Scripts

Ogni entità di Modus (elemento o associazione), ha una sua rappresentazione grafica. I dati relativi ad essa sono stati realizzati in un linguaggio di scripting definito da Enterprise Architect, e specificati nel profilo delle entità di cui alla Sezione C.1.3.

La Tabella 17 riporta il codice di un tipico Shape Script:

```
shape main
{
    h_Align = "center";
    println("«#STEREOTYPE#»");
    printwrapped("#NAME#");
    defSize(200, 50);

    startpath();
    moveto(30, 0);
```

```

lineto(0, 100);
lineto(70, 100);
lineto(100, 0);
lineto(30, 0);
endpath();
fillandstrokepath();
}

```

Tabella 17: Shape Script di un goal

La definizione è espressa in un linguaggio di programmazione C-like in cui vengono specificati, nell'ordine

- Informazioni riportate all'interno della forma, allineamento delle stesse e dimensioni dell'oggetto
- Coordinate degli estremi della figura piana identificata nel frame di contenimento

C.1.5 Modus Toolbox Profile

La creazione delle entità durante l'attività di Goal Modeling può avvenire, come descritto nella Sezione 3.3.2, direttamente dal project browser oppure contestualmente alla creazione di un diagramma, con il drag&drop di metaelementi e metassociazioni dalla toolbox di Modus Tool. La toolbox è stata definita nel profilo UML in Figura 100, Figura 101, Figura 102 e Figura 103.

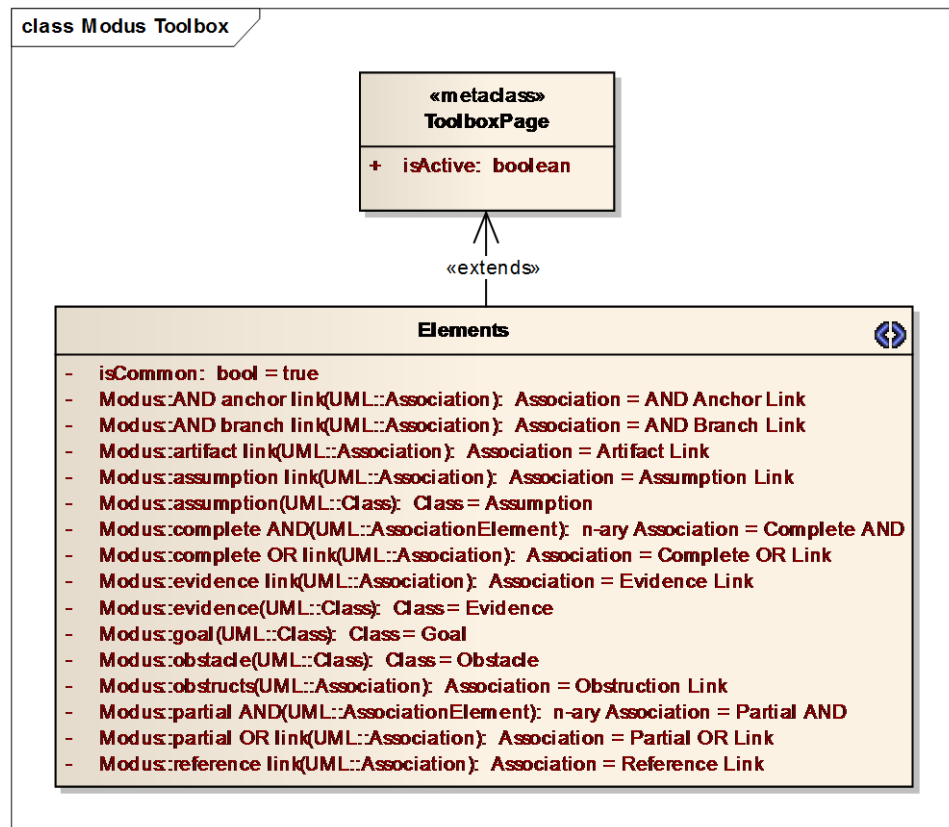


Figura 100: Profilo di definizione della toolbox di Modus Tool (elementi della toolbox)

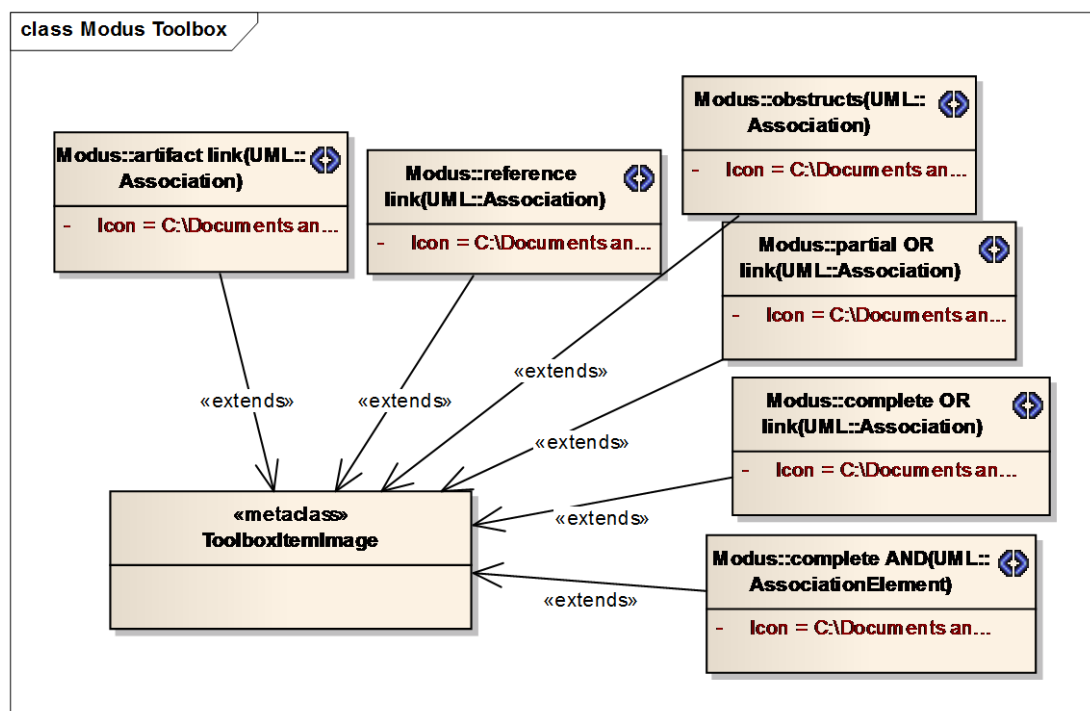


Figura 101: Profilo di definizione della toolbox di Modus Tool (immagini della toolbox, 1)

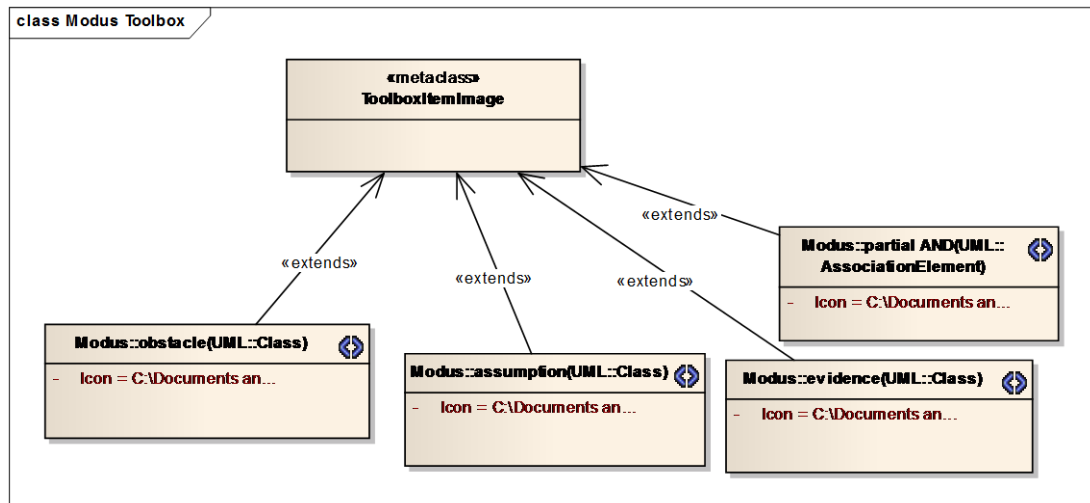


Figura 102: Profilo di definizione della toolbox di Modus Tool (immagini della toolbox, 2)

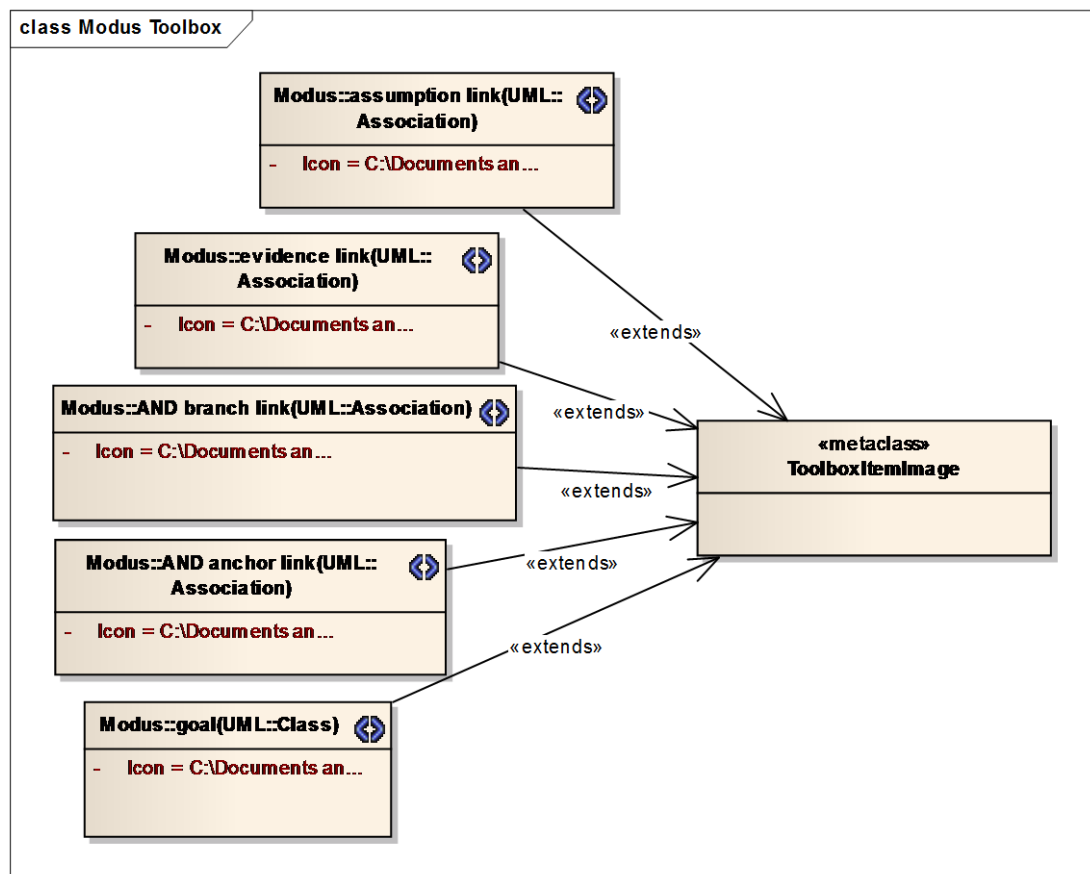


Figura 103: Profilo di definizione della toolbox di Modus Tool (immagini della toolbox, 3)

Lo stereotipo “Elements” contiene le definizioni di tutti gli elementi della toolbox, associando allo stereotipo UML dell’entità (Sezione C.1.3) una stringa di testo identificativa.

La metaclassa “ToolboxItemImage” è invece estesa dagli stereotipi relativi delle entità presenti nella toolbox, ognuno dei quali è decorato con l’attributo “Image” che specifica l’immagine bitmap dell’elemento, che in fase di esportazione del profilo viene convertita in formato Base64.

C.2 Model Validation

C.2.1 File delle regole di consistenza

Come già accennato nella Sezione 3.4.3, l’utilizzo del framework nativo di EA per la validazione delle regole, per quanto limitante sotto i citati aspetti, ha comunque consentito una sufficiente possibilità di interazione da parte dell’utente con la definizione delle regole.

All’avvio dell’add-in EA carica un file esterno in formato *Comma Separated Value (CSV)* contenente le definizioni delle regole di consistenza. L’utilizzo di un formato *human-readable* ha contribuito alla manutibilità del file stesso, che può pertanto essere modificato dall’utente con un semplice editor testuale prima della validazione.

Ogni riga del file CSV rappresenta una regola, ed ogni colonna rappresenta un attributo della regola stessa: la Tabella 18 riporta la lista degli attributi definiti per le regole di consistenza:

Colonna	Descrizione

Index	Indice numerico identificativo della regola
Description	Descrizione della regola
Error Message	Messaggio d'errore visualizzato dalla funzionalità di validazione di EA ogni volta che viene riscontrata un'infrazione della regola
Suggestion	Messaggio di suggerimento visualizzato nel report di validazione del tool ogni volta che viene riscontrata un'infrazione della regola
Subrules (opzionale)	Insieme di indici, separati da una virgola, delle eventuali sottoregole della regola

Tabella 18: Attributi delle regole di consistenza

Come definito nei requisiti (Sezione A.2), le regole sono divise in gruppi di appartenenza. Ogni gruppo, ad esclusione del generico “Other Rules”, è identificato da una speciale regola detta *abstract rule*, che rappresenta semanticamente la regola la cui semantica sintetizza quella delle sue sottoregole. Tutte e sole le abstract rules presentano un'insieme di indici delle regole di appartenenza al gruppo che identificano riportato nella colonna “Subrules” del file CSV.

Sebbene l'utente non possa intervenire sulla semantica delle singole regole (che è *hard-coded*), è comunque possibile rimuovere delle regole dalla lista a patto di mantenere la consistenza dei loro indici. La rimozione di una regola appartenente ad una abstract rule deve pertanto essere corredata dalla rimozione dell'indice della regola stessa dall'insieme “Subrules” della abstract rule.

La sintassi standard del file CSV prevede unicamente valori separati da un punto e virgola (“;”). È stata realizzata un'estensione della stessa con il supporto dei commenti *single-line*, anteposti da un doppio slash (“//”), propri di diversi linguaggi di programmazione. Utilizzando i commenti, l'utente può momentaneamente disabilitare il controllo di una regola senza tuttavia rimuoverla definitivamente dal file, senza pertanto la necessità di effettuare un backup del file delle regole.

C.2.2 Architettura: overview e sottosistemi

La Figura 104 riporta un’overview dei package interessati dalla feature di Model Validation. Come è possibile immaginare, l’utilizzo del framework offerto da EA per la validazione dei modelli ha consentito di focalizzare l’analisi sulla sola definizione del sottosistema di Model Validation.

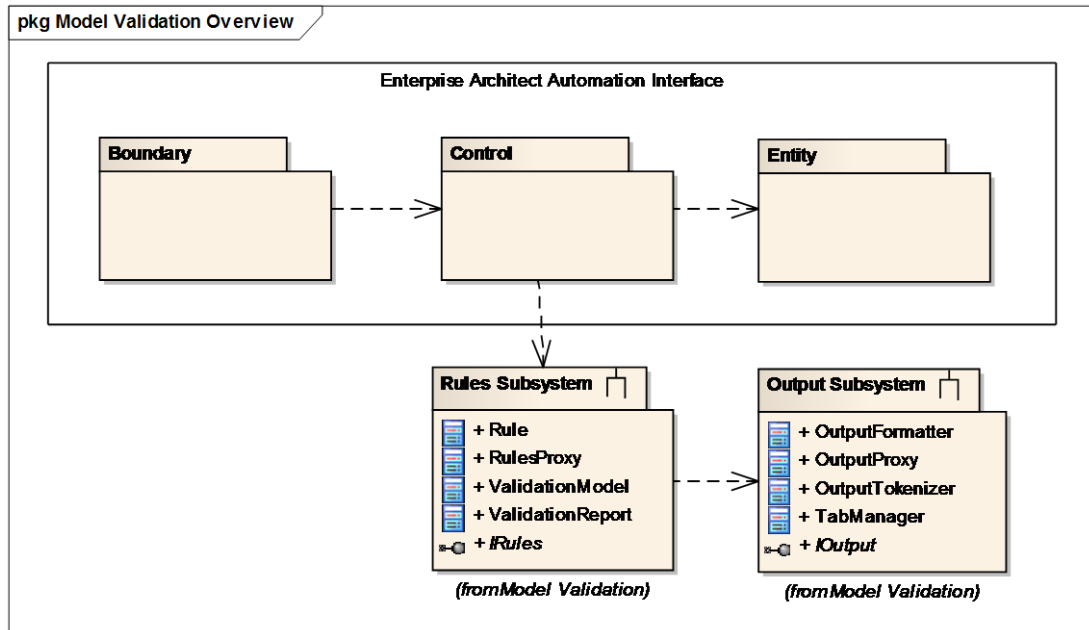


Figura 104: Package diagram della feature di Model Validation

I package “Boundary”, “Control” ed “Entity” relativi alla funzionalità di Model Validation sono definiti all’interno del modello di Enterprise Architect, pertanto ci si è semplicemente limitati a richiamare i metodi offerti dal sottosistema “Rules” all’interno degli *stub* delle classi control offerti dalla AI.

C.2.2.1 Sottosistema Rules

La Figura 105 riporta il dettaglio del sottosistema “Rules”, le cui classi sono brevemente discusse nella Tabella 19.

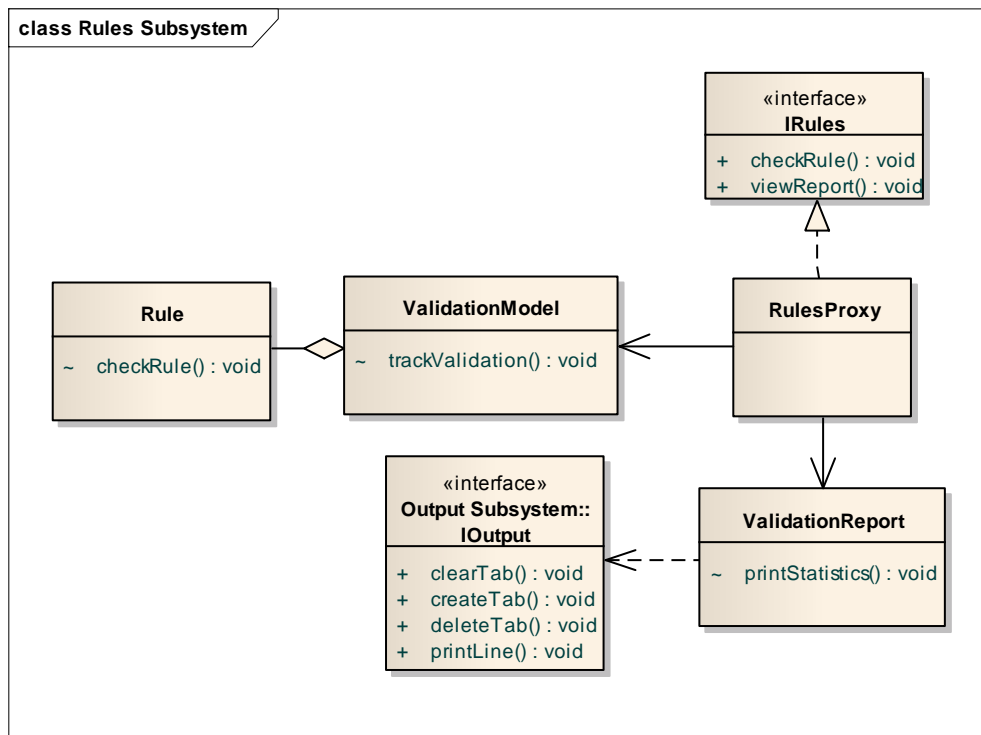


Figura 105: Class diagram del sottosistema “Rules”

Classe	Descrizione
RulesProxy	<p>Proxy del sottosistema. Realizza l’interfaccia <i>IRules</i> implementando i metodi</p> <ul style="list-style-type: none"> • <i>checkRule()</i>, che effettua il controllo di una regola in base all’indice della regola definito nel file esterno e all’oggetto selezionato • <i>viewReport()</i>, che visualizza il report relativo alla validazione
ValidationModel	<p>Rappresentazione di un processo di validazione, chiama i metodi di controllo delle regole e tiene traccia degli errori occorsi durante la stessa registrando gli eventi notificati</p>
Rule	<p>Rappresentazione di una regola di consistenza. Implementa il metodo di controllo della stessa, notificando al ValidationModel</p>

	informazioni sull'eventuale violazione della regola
ValidationReport	<p>Rappresentazione di un report di validazione. Raccoglie tutte le informazioni da presentare all'utente e le notifica utilizzando i servizi offerti dal sottosistema di gestione degli output. Ogni report viene visualizzato nella forma estesa, presentata contestualmente al controllo delle varie regole che intercorre durante la validazione, ed in forma compatta al termine della stessa:</p> <ul style="list-style-type: none"> • Report esteso: <ul style="list-style-type: none"> ○ Dettaglio del controllo di ogni regola su ogni elemento contenuto nel modello selezionato • Report compatto: <ul style="list-style-type: none"> ○ Tempo di validazione ○ Numero di regole controllate ○ Errori individuati, loro descrizione e suggerimenti per l'utente ○ Warning individuati, loro descrizione e suggerimenti per l'utente

Tabella 19: Descrizione delle classi del sottosistema "Rules"

Non si discuterà in questa sede della realizzazione delle regole di consistenza, comunque semplicemente riconducibile ad un task di implementazione nel modello a oggetti di EA di algoritmi di ricerca e visita su grafi.

C.2.2.2 Sottosistema Output

La Figura 106 riporta il dettaglio del sottosistema "Rules", le cui classi sono brevemente discusse nella Tabella 20.

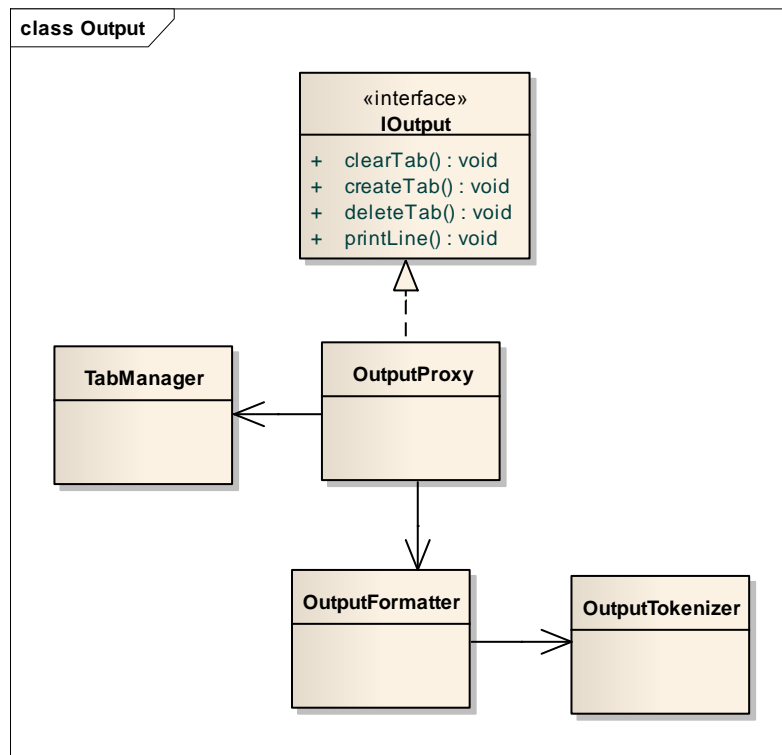


Figura 106: Class diagram del sottosistema “Output”

Classe	Descrizione
OutputProxy	<p>Proxy del sottosistema. Realizza l’interfaccia <i>IOutput</i> implementando i metodi</p> <ul style="list-style-type: none"> • <i>clearTab()</i>, che elimina il contenuto di una tab nella finestra “Output” di Enterprise Architect • <i>createTab()</i>, che crea una tab nella finestra “Output” di EA • <i>deleteTab()</i>, che elimina una tab dalla finestra “Output” di EA • <i>printLine()</i>, che crea una tab nella finestra “Output” di EA
OutputFormatter	<p>Gestore della formattazione di informazioni da visualizzare nella tab “Modus”. Implementa funzionalità di base quali tabulazione di stringhe, inserimento di separatori, collegamenti e rappresentazione testuale sintetica di tipi di dato complessi</p>
OutputTokenizer	<p>Gestore della tokenizzazione di informazioni da visualizzare nella tab</p>

	"Modus". Divide stringhe lunghe in stringhe più brevi, strutturandone il contenuto
TabManager	Gestore della tab "Modus" nella finestra "Output" di Enterprise Architect. Si occupa dell'interfacciamento del sottosistema con la Automation Interface, creando e visualizzando la tab e visualizzando in essa le informazioni

Tabella 20: Descrizione delle classi del sottosistema "Output"

C.3 Elicitation

C.3.1 Architettura: overview e sottosistemi

La Figura 107 riporta un'overview dei package interessati dalla feature di Elicitation, nei quali è visualizzato il dettaglio delle sole classi relative alla funzionalità.

La Tabella 21 riporta la descrizione delle classi di boundary relative alla feature di Elicitation.

Classe	Descrizione
ElicitProbabilities	<p>Finestra principale dell’add-in. Visualizza la lista degli utenti registrati nel repository di Enterprise Architect con un riepilogo generale, organizzato per aree di visualizzazione, di tutti i dati relativi all’attività di elicitazione delle opinioni:</p> <ul style="list-style-type: none"> • Lista delle sessioni del progetto corrente con il loro dettaglio: <ul style="list-style-type: none"> ○ Diagramma principale ○ Nome ○ Autore ○ Data creazione ○ Ultimo esperto che ha modificato la sessione ○ Ultima data di modifica della sessione ○ Status di consistenza • Snapshot dei diagrammi, lavagna e lista delle entità della sessione selezionata con il loro dettaglio: <ul style="list-style-type: none"> ○ Nome ○ Tipo ○ Diagramma di contenimento ○ Numero degli esperti che hanno risposto al questionario di elicitazione dell’entità ○ Status di elicitazione dell’utente corrente • Informazioni dettagliate relative all’entità selezionata: <ul style="list-style-type: none"> ○ Nome degli esperti che hanno risposto al questionario di elicitazione dell’entità ○ Risposte fornite al questionario da ogni esperto (opzionalmente su esplicita richiesta dell’utente)
ExportSession	<p>Finestra di export dei dati di una sessione. Consente all’utente di specificare:</p>

	<ul style="list-style-type: none"> • Path di salvataggio del file di export • Formato dei numeri decimali (<i>dot-separated</i> o <i>comma-separated</i>) • Esperti dei quali aggregare i dati
NewSession	<p>Finestra di creazione di una nuova sessione. Consente all'utente di specificare:</p> <ul style="list-style-type: none"> • Diagramma principale della sessione • Nome sessione • Tipo sessione
Questionnaire	<p>Finestra di visualizzazione del questionario di elicitazione. Contiene la lista delle domande relative all'entità cui esso si riferisce. Riporta inoltre il dettaglio dell'entità stessa già visualizzato nella finestra principale, al fine di evitare all'utente di dover affiancare le due finestre.</p>
SaveSession	<p>Finestra di salvataggio di una sessione. Consente all'utente di specificare la modalità di salvataggio della sessione:</p> <ul style="list-style-type: none"> • Salvataggio standard, con sovrascrittura della sessione modificata • Salvataggio con backup, mantenendo una copia di backup della sessione antecedente alle modifiche
SnapshotZoom	<p>Finestra di visualizzazione dello zoom di una snapshot di un diagramma. Contiene una versione ingrandita dell'immagine di un diagramma, disponendo della possibilità di ridimensionarla qualora sia più grande delle dimensioni di default della finestra.</p>

Tabella 21: Descrizione delle classi del package "Boundary"

C.3.1.2 Package Control

Nel package "Control" sono evidenziate le classi di controllo che partecipano alla realizzazione della feature di Elicitation.

In Figura 108 e Figura 109 è possibile notare le classi di controllo relative alle operazioni di *CRUD* delle sessioni, con esplicitata la loro dipendenza dal

sottosistema di persistenza "Data". Il meccanismo di scope dei dati delle sessioni descritto nella Sezione 3.5.3 è realizzato attraverso un'infrastruttura di classi di controllo la cui gerarchia di aggregazione rispecchia la gerarchia di contenimento degli scope relativi.

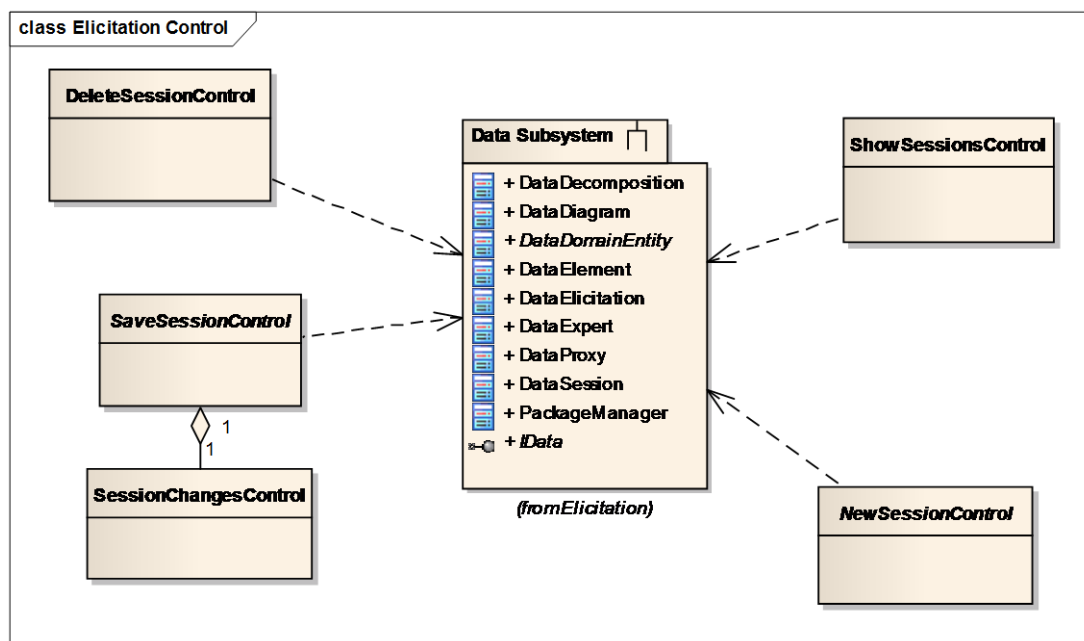


Figura 108: Class diagram del package "Control" della feature di Elicitation (1)

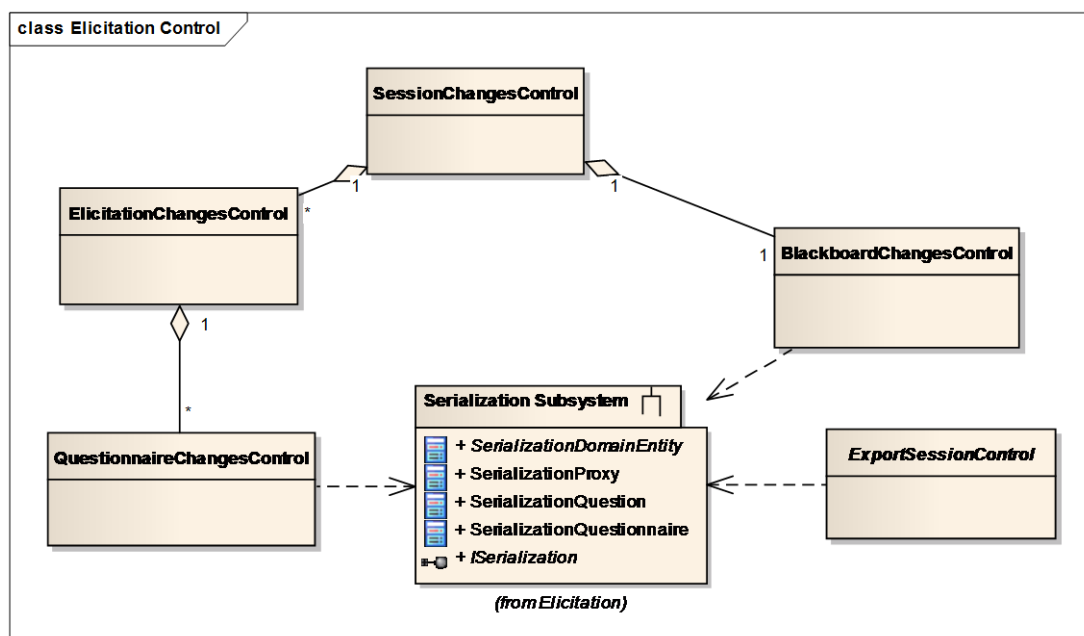


Figura 109: Class diagram del package "Control" della feature di Elicitation (2)

La Tabella 22 riporta una descrizione sintetica delle classi di controllo principali relative alla feature di Elicitation.

Classe	Descrizione
BlackboardChanges	Classe di controllo delle modifiche apportate dall'utente sulla lavagna (session scope). Memorizza temporaneamente (fino alla selezione di una nuova sessione) il contenuto della stessa.
DeleteSession	Classe di controllo della cancellazione di una sessione. Rimuove una sessione dal repository di Enterprise Architect ed aggiorna la lista delle sessioni visualizzate. Non è stato implementato alcun meccanismo di <i>recovery</i> di una sessione cancellata, pertanto l'operazione di cancellazione è da considerarsi irreversibile.
ElicitationChanges	Classe di controllo delle modifiche apportate dall'utente ai dati di elicitazione di un questionario (entity scope). Memorizza temporaneamente (fino alla selezione di una nuova sessione) le risposte confermate dall'utente.
ExportSession	Classe di controllo dell'export dei dati di una sessione. Coordina l'aggregazione dei dati relativi alle distribuzioni di probabilità e la loro propagazione all'interno del goal tree, serializzando i dati in un file CSV interpretabile da @RISK.
NewSession	Classe di controllo della creazione di una nuova sessione. Crea un nuovo artefatto stereotipato «session» all'interno del repository di EA, memorizzando le informazioni nei suoi tagged values.
QuestionnaireChanges	Classe di controllo delle modifiche apportate dall'utente ad un questionario durante la sua compilazione (elicitation scope). Valida sintatticamente in tempo reale le risposte inserite dall'utente, riportando un messaggio di errore qualora: <ul style="list-style-type: none"> • Vengano inseriti valori probabilità negative o minori di 1 • Vengano inseriti valori di probabilità nell'ordine errato • Non siano inserite le risposte di tutte le domande
SaveSession	Classe di controllo del salvataggio di una sessione. Lavora

	<p>secondo due modalità:</p> <ul style="list-style-type: none"> • Salvataggio standard (con sovrascrittura): aggiorna nel repository i dati relativi all'artefatto stereotipato «session» rappresentante la sessione da salvare, creando o aggiornando gli artefatti stereotipati «elicitation» che contengono i dati di elicitazione creati o modificate durante l'ultima sessione di lavoro • Salvataggio con backup: lascia inalterati i dati nel repository relativi alla sessione da salvare, creandone un nuovo clone che contiene tutte le modifiche apportate durante l'ultima sessione di lavoro
SessionChanges	Classe di controllo delle modifiche apportate dall'utente ad una sessione. Monitora le modifiche apportate alla lavagna (comunicando con BlackboardChangesControl) e le risposte dei questionari (comunicando con ElicitationChangesControl).
ShowSessions	Classe di controllo della visualizzazione delle informazioni di sessioni ed entità. Carica dal repository le informazioni da visualizzare e gestisce l'aggiornamento delle stesse a video.

Tabella 22: Descrizione delle classi del package "Control" della feature di Elicitation

C.3.1.3 Package Entity

Nel package "Entity" sono evidenziate le classi di entità che partecipano alla realizzazione della feature di Elicitation.

Alcune entità rappresentano dei *wrapper* di oggetti nel modello di Enterprise Architect, volti ad estenderne le responsabilità ed i dati incapsulati:

- Diagram incapsula EA.Diagram
- Element incapsula EA.Element
- Decomposition incapsula EA.Connector
- Expert incapsula EA.Client

Le altre invece rappresentano entità di dominio non esistenti nel modello di EA, e sono pertanto state definite in fase di analisi.

La Figura 110 riporta il class diagram del package “Entities”.

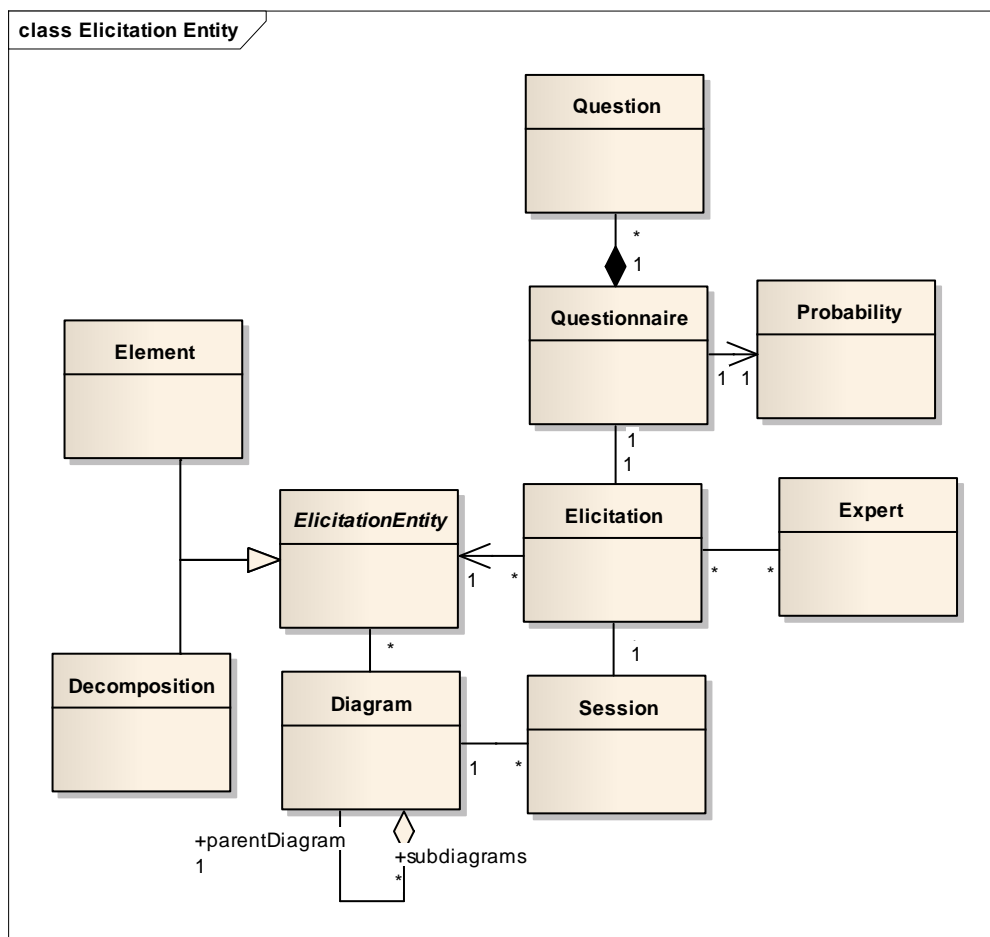


Figura 110: Class diagram del package "Entitiy" della feature di Elicitation

La Tabella 23 riporta una descrizione sintetica delle classi di controllo principali relative alla feature di Elicitation.

Classe	Descrizione
Decomposition	Rappresentazione di una decomposizione in un elemento di un goal diagram. Contiene il riferimento dell'Element decomposto, e la lista dei riferimenti degli Element decomponenti, nonché un

	riferimento alla <i>guid</i> dell'oggetto EA. Connector corrispondente nel repository.
Diagram	Rappresentazione di un goal diagram. Contiene un riferimento alla guid dell'oggetto EA. Diagram corrispondente nel repository, una lista dei Diagram in esso annidati e l'eventuale riferimento al Diagram padre.
Element	Rappresentazione di un elemento di un goal diagram. Contiene un riferimento alla guid dell'oggetto EA. Element corrispondente nel repository.
Elicitation	Rappresentazione dei dati di un'elicitazione di un'opinione di un esperto relativa ad un evento. Contiene i riferimenti dell'Expert che ha espresso l'opinione, dell'ElicitationEntity che rappresenta l'evento, della Session in cui l'elicitazione è avvenuta e del Questionnaire cui l'Expert ha risposto.
ElicitationEntity	Rappresentazione di un'entità della quale si desidera che un'esperto esprima un'opinione. Superclasse astratta di Element e Decomposition.
Expert	Rappresentazione di un esperto. Contiene un riferimento alla guid dell'oggetto EA. Client corrispondente nel repository.
Probability	Rappresentazione di una probabilità espressa sotto forma di distribuzione o valore puntuale.
Question	Rappresentazione di una domanda di un Questionnaire. Contiene id univoco della domanda, testo della stessa e risposta dell'utente .
Questionnaire	Rappresentazione di un questionario di elicitazione. Contiene la lista delle domande relative all'elicitazione dell'entità a cui si riferisce.
Session	Rappresentazione di una sessione di elicitazione. Contiene tutte le elicitazioni avvenute nel corso della sessione.

Tabella 23: Descrizione delle classi del package "Entity" della feature di Elicitation

C.3.1.4 Sottosistema Data

I dati persistenti della feature di Elicitation sono memorizzati all'interno del repository nel root model "Modus Data", la cui struttura è riportata nella Tabella 24.

Struttura	Descrizione
<ul style="list-style-type: none"> • Modus Data (root node) <ul style="list-style-type: none"> ○ Elicitations Data (package) <ul style="list-style-type: none"> ▪ Session 1 (folder) <ul style="list-style-type: none"> • «session»Session 1 (artifact) • «elicitation»Elicitation 1 (artifact) • ... • «elicitation»Elicitation n (artifact) ▪ ... ▪ Session n (folder) • ... • Root Model n 	<p>Root node contenente tutti i dati persistenti di Modus Tool</p> <p>Package contenente tutti i dati relativi alle attività di elicitazione del progetto corrente</p> <p>Folder contenente gli oggetti di decomposizione di goal e ostacoli, definito da Modus Tool</p> <p>Artifact stereotipato «session» contenente tutti i dati relativi ad una sessione</p> <p>Artifact stereotipato «elicitation» contenente tutti i dati relativi all'ecitazione di un'opinione di un esperto in una sessione relativamente ad un'entità di un diagramma</p>

Tabella 24: Struttura del root node Modus_Data

I dati sono organizzati all'interno degli oggetti del modello di EA attraverso l'utilizzo di tagged values, realizzando una memorizzazione dei dati che sia al contempo

human readable e *machine understandable*. Nonostante non sia mai necessario che l'utente interagisca direttamente e non attraverso le funzionalità del tool con i dati memorizzati, questa possibilità è comunque stata mantenuta per garantire supporto in caso di malfunzionamenti imprevisti. In casi estremi, infatti, è comunque possibile accedere ai dati dal menu "Properties > Tagged Values" degli oggetti di EA. Nessun test case ha comunque rilevato la necessità di utilizzare questa funzionalità, la cui esistenza è stata riportata per ragioni di esaustività.

La Tabella 25 riporta la lista dei tagged values degli stereotipi «elicitation» e «session» corredati da una breve descrizione.

Stereotipo	Tagged Value	Tipo	Descrizione
«elicitation»	Entity	GUID	GUID dell'entità che rappresenta l'evento dell'elicitazione
	Expert	GUID	GUID dell'esperto che ha espresso l'opinione sull'evento dell'elicitazione
	Questionnaire	XML	Documento XML contenente il questionario cui l'esperto ha risposto in fase di elicitazione
	Session	GUID	GUID dell'oggetto «session» che rappresenta la sessione di questa elicitazione
«session»	Author	GUID	GUID dell'utente che ha creato la sessione
	Blackboard	String	Testo della lavagna della sessione
	Created	Date	Data di creazione della sessione
	Diagram	GUID	GUID del diagramma principale della sessione
	Last Expert	GUID	GUID dell'ultimo esperto che ha salvato la sessione
	Modified	Date	Data dell'ultimo salvataggio della

sessione		
Type	String	Tipo della sessione

Tabella 25: Descrizione dei tagged values degli stereotipi di persistenza

La Figura 111 riporta il dettaglio del sottosistema “Data” in cui si è esplicitata la dipendenza della classe `DataDomainEntity` dal sottosistema “Serialization”.

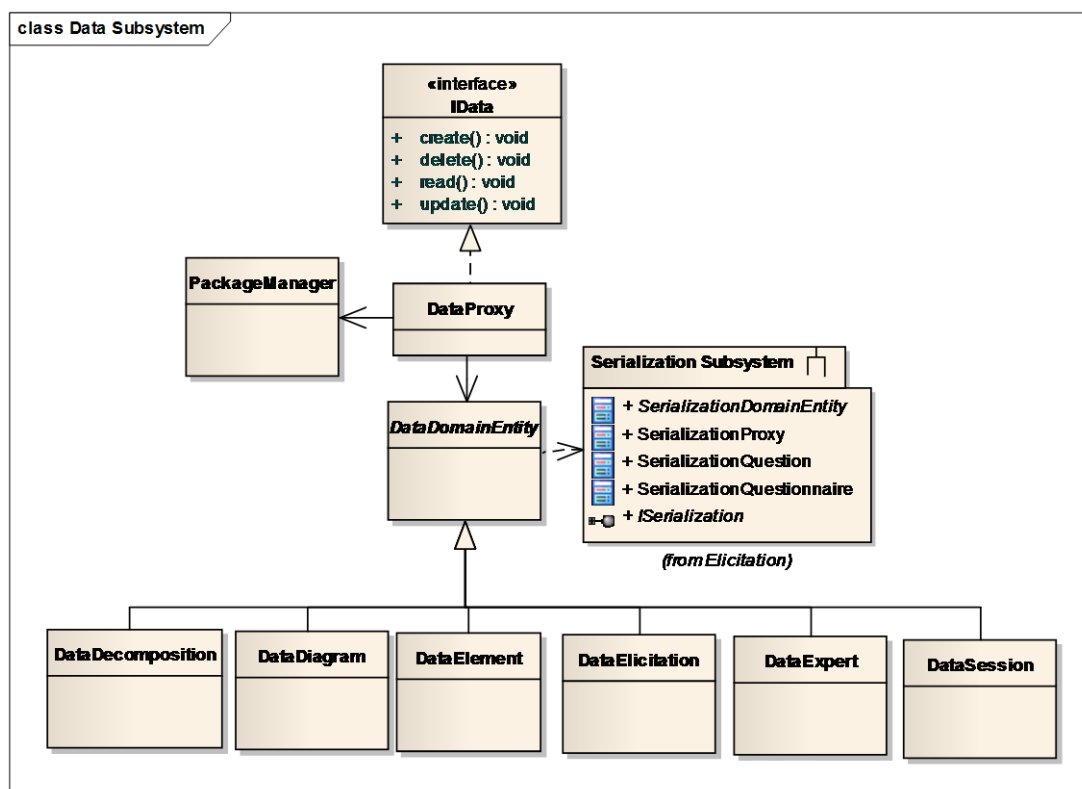


Figura 111: Class diagram del sottosistema “Data” della feature di Elicitation

Le classi del sottosistema “Data” sono brevemente discusse nella Tabella 26.

Classe	Descrizione
DataProxy	<p>Proxy del sottosistema. Realizza l'interfaccia <i>IData</i> implementando i metodi</p> <ul style="list-style-type: none"> • <i>create()</i>, che crea un nuovo oggetto nel repository • <i>delete()</i>, che elimina permanentemente un oggetto dal repository

	<ul style="list-style-type: none"> • <i>read()</i>, che legge i dati di un oggetto dal repository • <i>update()</i>, che aggiorna i dati di un oggetto nel repository
PackageManager	Gestore della rappresentazione logica dei dati nel repository. Monitora ed assicura la consistenza del root node Modus Data che contiene tutte le informazioni relative di sessioni ed elicitazioni, ripristinando automaticamente il <i>file system</i> interno di Modus Tool qualora vengano riscontrate inconsistenze nei dati
DataDomainEntity	Gestore della persistenza delle entità di dominio nel repository. Classe astratta che fornisce funzionalità di base come <i>caching</i> degli elementi letti, criteri di ricerca basati su identificativi locali (definiti all'interno di Modus Tool) e globali (definiti all'interno del repository) e <i>marshalling/unmarshalling</i> delle proprietà degli oggetti nei tagged values.
DataDiagram	Gestore della persistenza di oggetti di tipo Diagram. Legge dal repository informazioni sui goal diagram (EA. Diagram).
DataDecomposition	Gestore della persistenza di oggetti di tipo Decomposition. Legge dal repository informazioni sui connettori (EA. Connector) degli elementi di un diagramma.
DataElement	Gestore della persistenza di oggetti di tipo Element. Legge dal repository informazioni sugli elementi (EA. Element) di un diagramma.
DataElicitation	Gestore della persistenza di oggetti di tipo Elicitation. Crea, legge, aggiorna ed elimina dal repository informazioni sulle elicitazioni. Ogni elicitazione è memorizzata nel repository come un EA. Element di tipo Artifact stereotipato «elicitation» all'interno del folder della propria sessione di appartenenza. Le informazioni delle elicitazioni sono memorizzate nell'oggetto «elicitation» in forma testuale all'interno di appositi tagged values.
DataExpert	Gestore della persistenza di oggetti di tipo Expert. Legge dal repository informazioni sugli esperti (EA. Client).

DataSession	Gestore della persistenza di oggetti di tipo Session. Crea, legge, aggiorna ed elimina dal repository informazioni sulle sessioni di elicitazione. Ogni sessione è memorizzata nel repository come un EA. Element di tipo Artifact stereotipato «session» all'interno di un folder nel package "Elicitations" con lo stesso nome della sessione. Le informazioni della sessione sono memorizzate in forma testuale all'interno di appositi tagged values dell'oggetto «session», mentre le elicitazioni sono memorizzate in oggetti «elicitation» contenuti nello stesso folder.
--------------------	--

Tabella 26: Descrizione delle classi del sottosistema "Data" della feature di Elicitation

C.3.1.5 Sottosistema Serialization

L'utilizzo di un layer di persistenza che memorizza i propri dati in forma testuale all'interno di tagged values ha richiesto un meccanismo di serializzazione che fosse in grado di codificare e decodificare gli oggetti da salvare persistentemente.

Come avviene di norma in questi casi, si è scelto utilizzare lo standard XML [37] per la rappresentazione di dati strutturati. L'utilizzo di un linguaggio di definizione strutturale ha notevolmente semplificato le operazioni marshaling ed unmarshaling dei dati, che si sono potute basare su *parser* evoluti già disponibili, pur mantenendo un considerevole grado di flessibilità garantita dall'estensibilità propria del linguaggio stesso. Evitare l'utilizzo di un formato ad-hoc di rappresentazione dei dati ha inoltre notevolmente ridotto il tempo di testing delle funzionalità del sottosistema.

Per brevità non si riportano documenti tipo e schemi XML relativi alla memorizzazione dei dati, ma ci si limita a menzionare l'uso di tale meccanismo per le sole classi Question e Questionnaire. Tutti i dati non strutturati sono infatti stati memorizzati direttamente nei tagged values degli oggetti del repository, delineando pertanto un approccio di serializzazione dei dati ibrido. L'organizzazione degli stessi è infatti definita attraverso il meccanismo nativo dei tagged values UML per tipi di dato semplici, ed attraverso documenti XML per tipi di dato strutturati.

La Figura 112 riporta il dettaglio del sottosistema “Serialization”, le cui classi sono brevemente discusse nella Tabella 27.

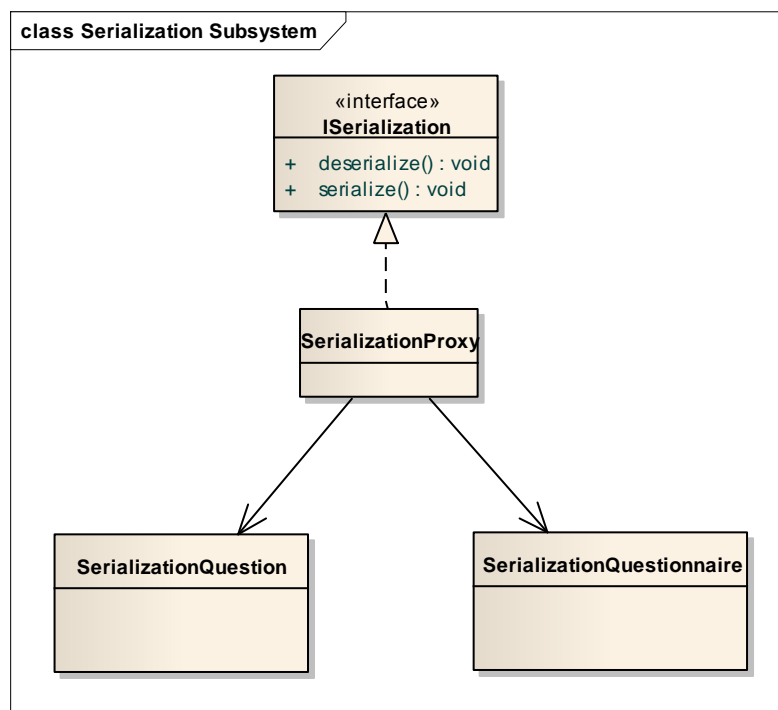


Figura 112: Class diagram del sottosistema “Serialization” della feature di Elicitation

Classe	Descrizione
SerializationProxy	<p>Proxy del sottosistema. Realizza l’interfaccia <i>ISerialization</i> implementando i metodi</p> <ul style="list-style-type: none"> • <i>deserialize()</i>, che deserializza un un documento XML creando un oggetto le cui informazioni di stato sono rappresentate nel documento • <i>serialize()</i>, che serializza un oggetto rappresentando il suo stato in un documento XML
SerializationQuestion	<p>Gestore della serializzazione e deserializzazione di un oggetto di tipo Question. Memorizza id, testo della domanda e testo della risposta in un un documento XML.</p>
SerializationQuestionnaire	<p>Gestore della serializzazione e deserializzazione di un oggetto di tipo Questionnaire. Memorizza il tipo dell’entità</p>

cui il questionario si riferisce e la lista degli oggetti di tipo
Question che lo compongono

Tabella 27: Descrizione delle classi del sottosistema “Serialization” della feature di Elicitation