

Il Comune di Roma, d'intesa con il consorzio Metrebus, ha avviato nel 2005 l'erogazione del servizio di *car sharing* in città¹ (chiamato RomaCarSharing). Il servizio di car sharing permette ai soci di noleggiare, presso uno dei garage convenzionati, disseminati nel territorio, un'autovettura anche solo per poche ore, pagando l'uso effettivo del veicolo (ovvero un costo fisso, più un costo orario, mentre il carburante è compreso). I vantaggi sono molteplici, soprattutto per coloro che non usano frequentemente l'auto, potendo evitare i costi (assicurazione, bollo, manutenzione) derivanti dal possesso di un'auto privata.

Si vuole progettare un'applicazione che permetta di gestire alcune informazioni sul servizio di car sharing, relativamente ai soci e alle auto nolggiate.

Si richiede di effettuare la fase di Analisi, modellando la seguente specifica dei requisiti.

Requisiti

Le società possono stipulare delle convenzioni con il servizio, che permettono di ottenere sconti sui noleggi. In particolare, le convenzioni (di cui interessa il nome e il tasso di sconto applicato), si dividono in due categorie: quelle che incentivano il noleggio di auto ecocompatibili, e quelle che invece applicano lo sconto sui noleggi delle auto di alcune categorie (almeno una).

Il sistema deve offrire le seguenti funzionalità aggiuntive:

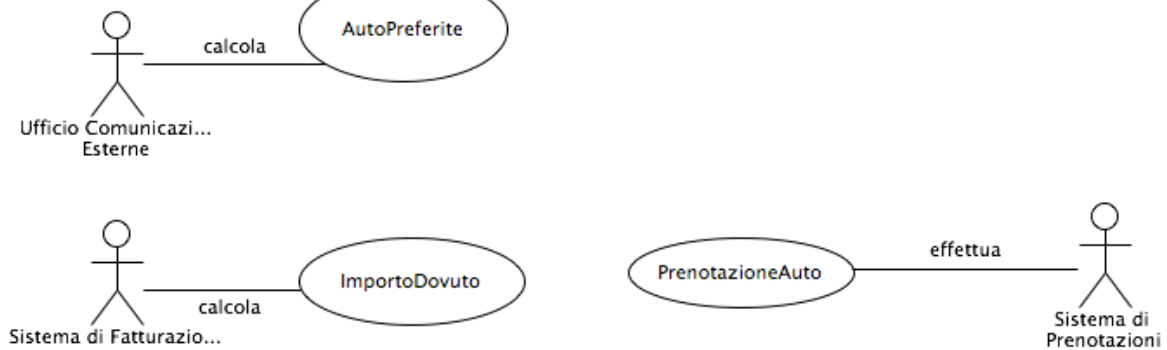
1. Il sistema di fatturazione, data una categoria di auto, vuole conoscere le società che hanno le convenzioni più vantaggiose (ovvero con tassi di sconto più elevati) per auto di quella categoria;
2. Il sistema di fatturazione, dato un insieme di società, vuole conoscere quelle che hanno stipulato con il servizio RomaCarSharing almeno una convenzione per noleggiare auto ecocompatibili.

¹Per ora, in via sperimentale, solo nel III Municipio, dal 2006 sarà gradualmente esteso a tutto il territorio cittadino. Per info: www.comune.roma.it.

1 Fase di Analisi

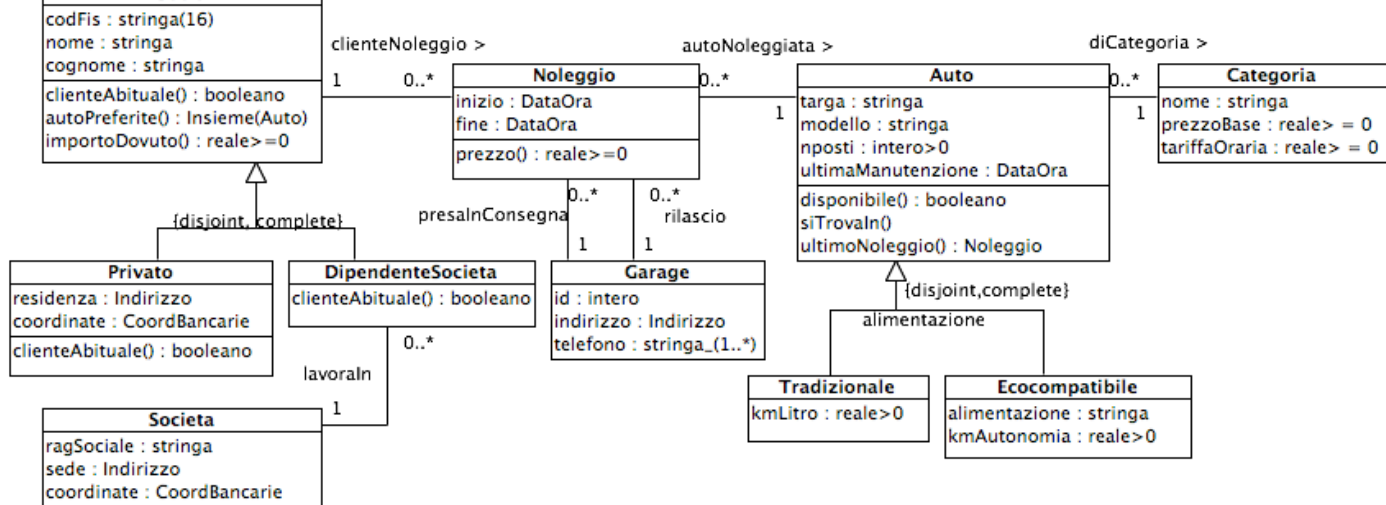
1.1 Diagramma degli Use Case prodotto nel passo A.1

Visual Paradigm for UML Community Edition [not for commercial use]



1.2 Diagramma delle classi UML prodotto nel passo A.1

Visual Paradigm for UML Community Edition [not for commercial use]



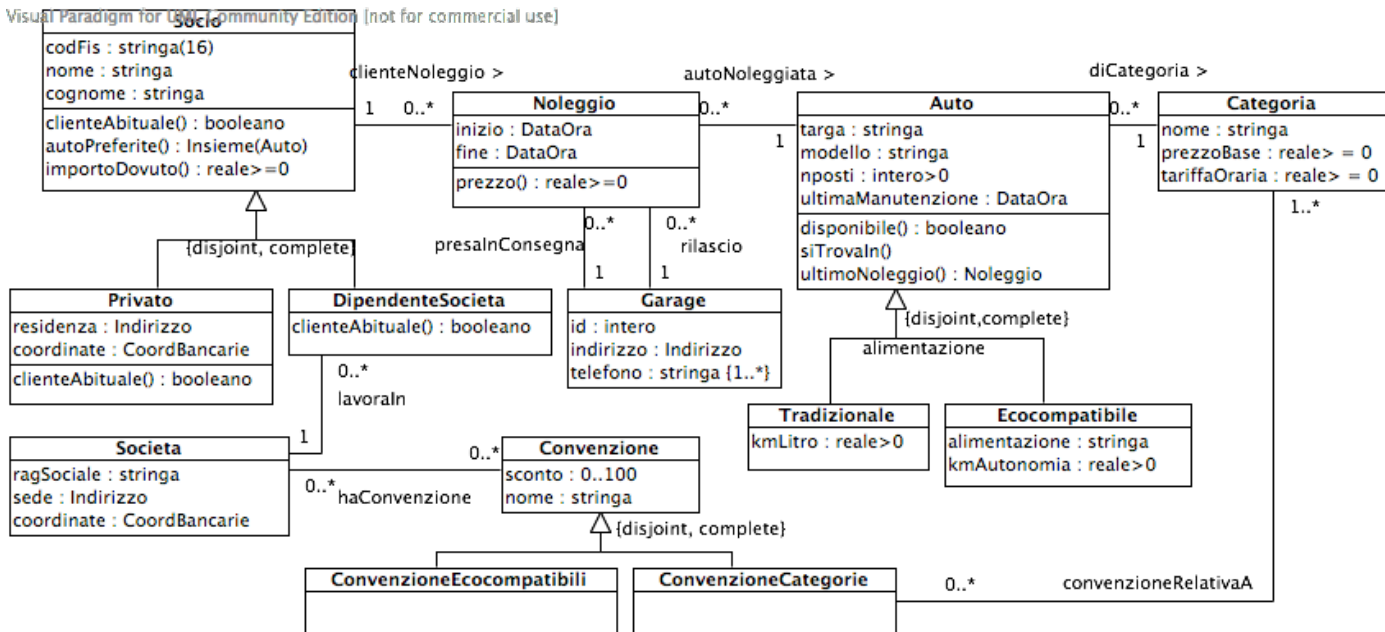
1.3 Diagramma degli Use Case

Visual Paradigm for UML Community Edition (not for commercial use)



1.4 Diagramma delle classi UML

Visual Paradigm for UML Community Edition (not for commercial use)



1.5 Specifica degli use case

Use case VerificheSuConvenzioni

SpecificaUseCase VerificheSuConvenzioni

societaConConvPiuVantaggiose(c : Categoria) : Insieme(Societa)
pre: nessuna
post: Detto c.convenzioneRelativaA l'insieme dei link di tipo
convenzioneRelativaA in cui 'c' e' coinvolto, sia

$$CC = \{cc \in \text{ConvenzioneCategorie} \mid \langle cc, c \rangle \in \text{convRelativaA}\}$$

l'insieme dei relativi oggetti di classe ConvenzioneCategorie.

Sia inoltre 'CCmax' il sottoinsieme degli oggetti di classe
'ConvenzioneCategorie' in CC che hanno il valore piu' alto
per l'attributo 'sconto', ovvero:

$$CC_{max} = \left\{ cc \in CC \mid \text{per ogni altro oggetto } cc' \in CC \text{ con } cc' \neq cc \text{ si ha che } \right. \\ \left. cc.sconto \geq cc'.sconto \right\}.$$

result e' pari all'insieme delle societa' coinvolte in link di tipo
'haConvenzione' con oggetti in CCmax, ovvero:

$$\text{result} = \{s \in \text{Societa} \mid \text{esiste } cc \in CC_{max} \text{ t.c. } \langle s, cc \rangle \in \text{haConvenzione}\}$$

almenoUnaConvenzioneEco(I : Insieme(Societa)) : Insieme(Societa)

pre: nessuna
post: 'result' e' il sottoinsieme di 'I' di societa' coinvolte con almeno un link di
tipo 'haConvenzione' ad oggetti di classe 'ConvenzioneEcocompatibili',
ovvero:

$$\text{result} = \{ s \text{ in 'I' } \mid \text{esiste 'ce' in ConvenzioneEcocomp. t.c.} \\ \langle s, ce \rangle \text{ in haConvenzione } \}$$

FineSpecifica