Patró **Proxy**

Estem dissenyant una classe **Subjecte** en la que s'executen operacions que empren serveis que:

- No es troben dins l'àmbit de la nostra aplicació, o
- Son costosos de crear, i no volem que la nostra aplicació assumeixi la responsabilitat de crear-los, o
- Estan subjectes a una política d'accessos o de permisos diferent, o
- Es vol incloure més seguretat o més control sobre un servei extern, (per ex: controlar l'ús que es fa de l'objecte, implementar una estadística, controlar bloqueigos d'altres parts quan es fa ús de l'objecte, etc.)

En aquests casos, es crea una classe anomenada **Proxy**, que especialitza la classe subjecte, i que es situa entre la classe que precisa del servei, i la classe real que ofereix el servei, mantenint una referència amb aquesta última. La classe Proxy implementa una interfície, que es la mateixa que la de la classe real. L'aplicació fa servir aquesta interfície, i Proxy implementa tot el necessari per fer servir el servei real, desacoblant la classe subjecte de les responsabilitats d'aquest servei. A més, Proxy pot implementar altra funcionalitat de control o de seguretat, i pot emmagatzemar altra informació necessària i complementaria a la del servei al que supleix.

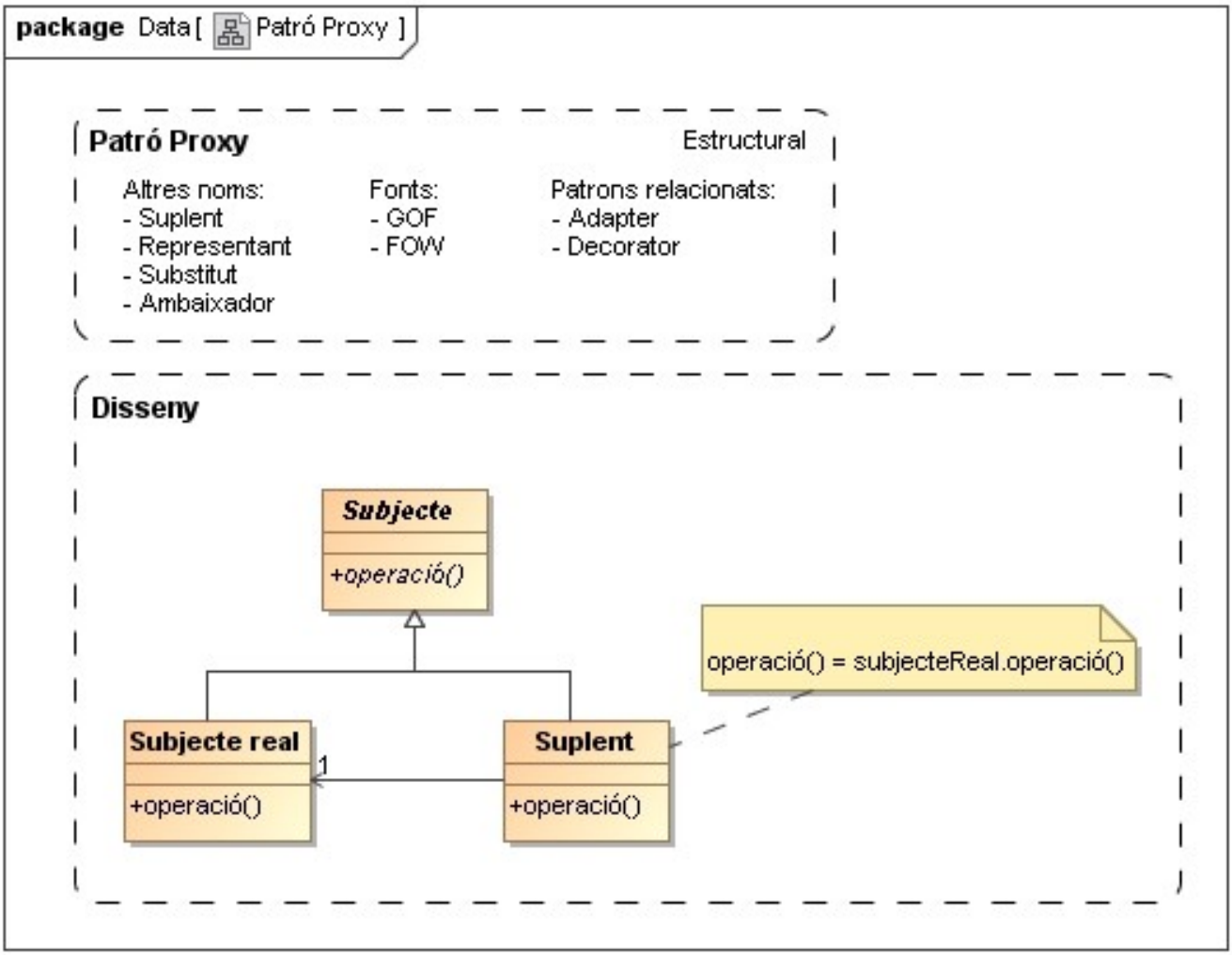
Característiques

Nom del patró	**Tipus de patró**	**Fonts documentals**
Proxy	Estructural	GOF FOW
Altres noms	**Patrons relacionats**	
Substitut Representant Suplent Surrogate Ambaixador	Adapter Decorator	

El patró **Adapter** proposa una interfície diferent per al servei real al qual adapta, mentre que Proxy implementa en la seva interfície els mateixos mètodes que el servei real, (o el subconjunt útil d'aquests mètodes per la classe Subjecte). Proxy es transparent en el seu ús al servei al que substitueix, si no és que opcionalment pot implementar mètodes per a garantir la seguretat, o establir controls; mentre que Adapter proposa canvis de funcionament respecte al servei al que adapta.

El patró **Decorator** afegeix noves responsabilitats, (proporciona nova funcionalitat emprant la herència), sobre una classe concreta. No representa un servei.

Disseny

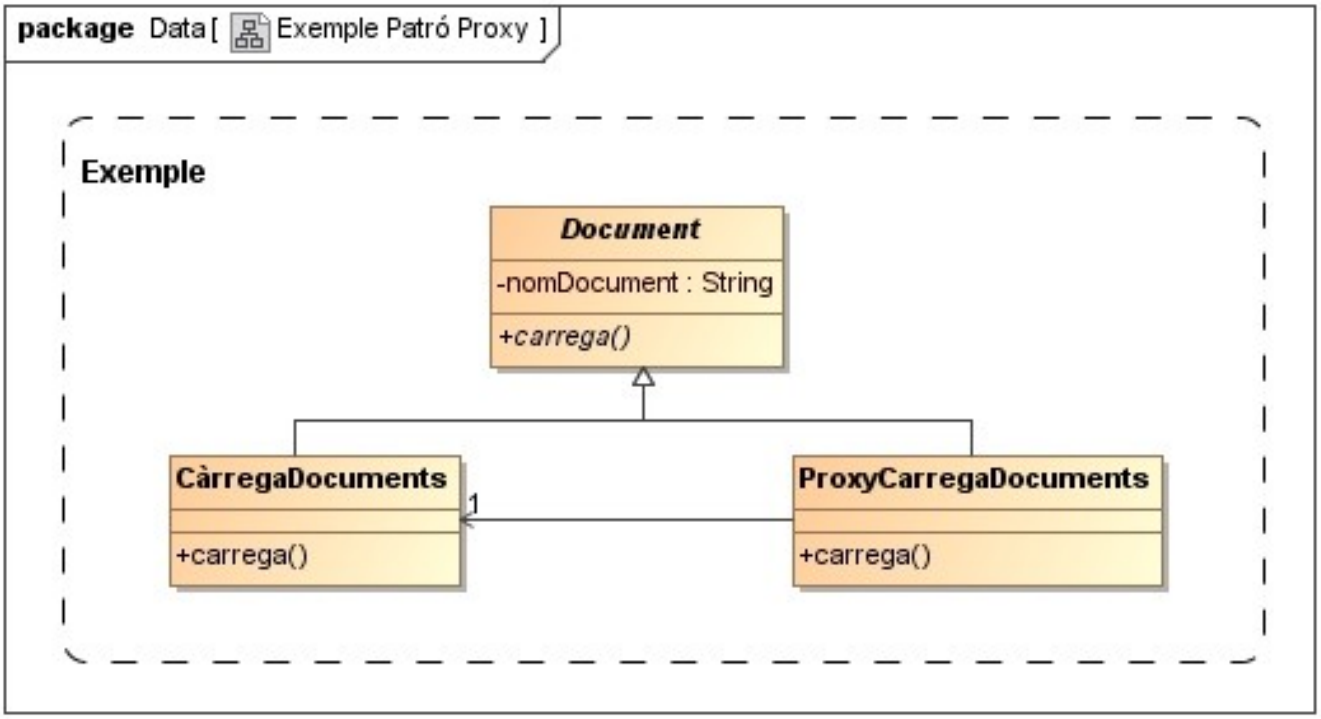


Subjecte
Es la classe de l'aplicació en desenvolupament que precisa d'un servei que s'implementarà amb Proxy.

Subjecte real
Aquesta classe es correspon a la classe que proporciona el servei, i de la qual no volem responsabilitzar-nos de la seva complexitat, o de la que no volem que l'aplicació en conegui els detalls. Proxy treballa per resoldre això.

Suplent, (o Proxy)
La classe proxy, (o suplent), crea una interfície que treballa que emula la implementació de la classe real. La classe Subjecte fa ús d'aquesta interfície enlloc de fer servir directament el subjecte real.

Exemple



En l'exemple tenim una aplicació que gestiona documents, representada per la classe **Document**. Una de les responsabilitats de la classe Document és la càrrega de documents. Suposem que la càrrega és diferent depenent del format de l'arxiu i altres característiques. No volem que la classe Document es responsabilitzi de totes les vicissituds de la càrrega de documents, així que fa ús de la classe **CàrregaDocuments**. Però aquesta classe és conforma com un servei, del qual no coneixem la seva ubicació exacta i/o no som responsables de la seva ubicació ni evolució. Així, fem ús d'una classe, (aquesta si, nostra), que implementa la mateixa interfície de funcionament que CàrregaDocuments, que és **ProxyCarregaDocuments**. L'aplicació fa ús del proxy, el qual s'encarrega de contactar amb el servei i fer-lo servir de forma adequada, i de forma independent a la resta de l'aplicació.