

Przy ocenie narzędzia do modelowania UML można stosować następujące kryteria. Istnieje wiele produktów spełniających je, a ponieważ chcesz, aby wszystko było jak najlepsze, poświęć czas, aby ocenić produkty testowane na podstawie tej listy.

Wsparcie dla repozytoriów

W przypadku dużego projektu repozytorium jest niezbędne do współdzielenia projektów komponentów między programistami. Dwóch lub więcej programistów może udostępniać komponenty modelu lub nawet współpracować nad rozwojem pojedynczego komponentu, określając prawa własności i udostępniania na odpowiednim poziomie.

Repozytorium jest zwykle budowane na bazie danych, która zapewnia funkcje udostępniania danych i kontroli współbieżności. Zapewniając dostęp blokujący i umożliwiający tylko odczyt, repozytorium pozwala jednemu deweloperowi na posiadanie modelu, umożliwiając innym odczytywanie modelu i jego komponentów, a także włączenie tych komponentów do własnych projektów.

Ważne: Narzędzie powinno umożliwiać pobieranie tylko tych komponentów, które są potrzebne z innego modelu, bez konieczności importowania całego modelu.

Innym interesującym sposobem na zbudowanie repozytorium jest kod źródłowy projektu, wykorzystujący system kontroli kodu źródłowego w celu zapewnienia kontroli współbieżności. Zaletą tego podejścia jest wyższy stopień synchronizacji między kodem a modelem. Kolejną korzyścią jest eliminacja kolejnego źródła danych.

Dzięki narzędziom do modelowania, które obsługują repozytorium, zmiany w dowolnym komponencie powinny być automatycznie propagowane do dowolnego projektu, który importuje komponent.

Round-Trip Engineering (inżynieria w obie strony)

Możliwość przesyłania kodu źródłowego do przodu i do tyłu (Java, C++, CORBA IDL) jest złożonym wymaganiem, które zapewniają sprzedawcy z różnym powodzeniem. Pomyślnie połączenie tych dwóch funkcji, inżynierii normalnej i inżynierii odwrotnej, definiuje się jako inżynierię w obie strony.

Inżynieria normalna jest bardzo przydatna przy pierwszym generowaniu kodu z modelu. Dzięki temu zaoszczędzisz wiele przyziemnej pracy związanej z klasami, atrybutami i metodami.

Inżynieria odwrotna jest bardzo przydatna zarówno do przekształcania kodu w model, gdy wcześniej nie istniał model, jak i do resynchronizacji modelu z kodem na końcu iteracji.

Podczas iteracyjnego cyklu rozwoju, po zaktualizowaniu modelu w ramach iteracji, kolejny etap inżynierii normalnej powinien pozwolić na odświeżenie kodu z dowolnymi nowymi klasami, metodami lub atrybutami, które zostały dodane do modelu. Ten krok jest mniej powszechnie stosowany przez programistów, ponieważ wiele narzędzi może beznadziejnie łączyć kod źródłowy w procesie. Problem polega na tym, że kod źródłowy zawiera znacznie więcej niż model; narzędzia muszą być bardzo biegłe w rekonstruowaniu kodu źródłowego, który istniał przed nowym etapem inżynierii normalnej.

Narzędzie do modelowania powinno co najmniej z powodzeniem wspierać inżynierię normalną po raz pierwszy i inżynierię wsteczną w całym procesie. Ponadto narzędzie nie powinno mieć problemów z inżynierią wsteczną pełnego języka Java. Upewnij się, że zweryfikujesz tę funkcję dla własnego kodu źródłowego, ponieważ zaobserwowaliśmy, że dobrze znane narzędzia zawiodą przy podstawowych operacjach, takich jak generowanie klas wewnętrznych. Utkniesz wykomementowując zbędny kod – bardzo bolesny proces.

Dokumentacja HTML

Narzędzie do modelowania obiektów powinno zapewniać płynne generowanie dokumentacji HTML dla modelu obiektowego i jego komponentów. Dokumentacja HTML zapewnia statyczny widok modelu obiektowego, dzięki któremu każdy programista korzystający z modelu może szybko odnieść się do przeglądarki, bez potrzeby uruchamiania samego narzędzia do modelowania. Ponadto, tworząc dokumentację HTML, liczba wymaganych licencji na narzędzie do modelowania może zostać zmniejszona o liczbę osób, które potrzebują dostępu do danych modelu tylko do odczytu.

Dokumentacja HTML powinna zawierać obraz bitmapy każdego z diagramów w modelu i powinna zapewniać nawigację w całym modelu za pomocą hiperłączy. Ilość czasu potrzebnego do wygenerowania kodu HTML powinna być rozsądna. Wiele produktów dostępnych dzisiaj obsługuje te funkcje z różnym skutkiem. Ponownie, upewnij się, że przetestujesz tę funkcję samodzielnie.

Pełna obsługa UML 1.3

Podczas gdy wiele narzędzi zapewnia pełne wsparcie dla UML 1.3, w rzeczywistości jest to złożone wymaganie, a niektóre narzędzia mogą nie spełniać reklamowanych roszczeń dotyczących pełnego wsparcia. Jako minimum schematy, które powinny być obsługiwane, to Diagramy przypadków użycia, klasy, współpracy, sekwencji, paczek i stanów.

Listy wyboru dla klas i metod

Narzędzie do modelowania powinno zapewniać listy wyboru w kilku kluczowych interfejsach:

Schematy współpracy i sekwencji - Narzędzie powinno umożliwiać przypisanie obiektu do klasy z listy klas w modelu. Powinno umożliwiać wybieranie komunikatów wysyłanych między obiektami z poprawnej listy metod dla obiektu (klasy), która odbiera komunikat.

Diagram klasowy - narzędzie powinno umożliwiać import klas z innych pakietów lub modeli w oparciu o wybór klasy z listy klas w pakiecie.

Funkcja listy wyboru znacząco przyczynia się do intuicyjności narzędzia do modelowania i może być uważana za niezbędną funkcję. Opracowanie diagramów sekwencji i współpracy jest znacznie łatwiejsze dzięki możliwości szybkiego wyboru wiadomości, które chcesz wysłać z jednego obiektu do drugiego.

Integracja modelowania danych

Narzędzie do modelowania obiektów powinno umożliwiać integrację z obiektami do modelowania danych. Istnieje wiele sposobów na zapewnienie tej funkcjonalności. Jednym ze sposobów jest narzędzie UML udostępniające funkcję umożliwiającą przekształcenie modelu obiektowego w DDL, który jest SQL niezbędny do tworzenia tabel dla klas. Innym sposobem jest narzędzie UML do eksportowania metadanych do narzędzia do modelowania danych, które może importować metadane i wykorzystywać je jako podstawę dla modelu danych. Zaawansowany, zintegrowany zestaw narzędzi powinien umożliwiać synchronizację modeli danych i modeli obiektów po każdej iteracji projektu.

Wersjonowanie

Narzędzie do modelowania powinno zezwalać na zapisywanie wersji, aby podczas kolejnych iteracji dostępna była poprzednia wersja do przywracania lub do zachowania istniejącego kodu, który opiera się na tej wersji.

Nawigacja modelu

Narzędzie do modelowania powinno zapewnić silne wsparcie nawigacyjne, aby umożliwić programistom nawigowanie po wszystkich diagramach i klasach w modelu. Katalog lub lista wyboru klas posortowanych według nazwy jest jednym ze sposobów umożliwienia projektantowi przejścia do żądanej klasy na diagramie.

W przypadku dużych diagramów narzędzie powinno zapewniać łatwość nawigacji podczas powiększania i przesuwania.

Narzędzie powinno również umożliwiać łatwą nawigację do kodu źródłowego dla klasy, gdy używana jest inżynieria w obie strony.

Drukowanie

Narzędzie do modelowania powinno umożliwiać tworzenie dokładnych interpretacji dużych diagramów za pomocą drukowania wielostronicowego. Podgląd wydruku i skalowanie powinny być

obsługiwane, aby umożliwić łatwe dopasowanie diagramu do żądanej liczby stron. Umiejętność dopasowania diagramu do pojedynczej strony znajduje się wysoko na tej liście. Niestety, okazuje się, że wiele narzędzi ma problemy z wykonywaniem tego ważnego zadania w poprawny sposób.

Widoki diagramu

Narzędzie do modelowania powinno ułatwić dostosowanie widoku klasy i jej szczegółów. Na przykład, powinno być możliwe wykluczenie wszystkich metod pobierania / ustawiania z diagramu, ponieważ mają tendencję do bałaganienia, a nie do wyjaśnienia diagramu. Pełny opis metod powinien być łatwo widoczny lub ukryty, w zależności od pożądanego poziomu szczegółowości. Widoczność atrybutów i metod (prywatna, chroniona, publiczna) powinna być kolejnym kryterium używanym do wyboru tego, co pokazać lub ukryć na diagramie.

Eksportowanie diagramów

Jedną z kluczowych cech, która często jest pomijana, jest możliwość eksportu schematów do formatu, który można zaimportować do dokumentu edytora tekstu lub strony internetowej. Najpopularniejsze formaty graficzne używane do eksportu to GIF, PNG i JPEG. Podczas eksportowania narzędzie powinno umożliwiać określenie preferowanej rozdzielczości i rozmiaru tworzonej grafiki.

Tworzenie skryptów

Skrypty to kolejna potężna funkcja, którą powinno wspierać narzędzie do modelowania. Za pomocą narzędzie do tworzenia skryptów użytkownik zaawansowany może tworzyć skrypty, które mogą bezpośrednio uzyskać dostęp do modelu obiektu w narzędziu do modelowania w celu tworzenia innych artefaktów. Przykładami tych artefaktów są arkusze kalkulacyjne do zarządzania projektami dla tworzonego projektu, spersonalizowana dokumentacja, dostosowany kod, raporty i metryki. Przykładem dostosowywanego kodu są kolekcje klas i metody get / set używane do uzyskiwania dostępu do zasobów tych klas.

Odporność

Chcesz niezawodne narzędzie UML, aby zapobiec utracie przez użytkowników potencjalnej godziny produktywności, gdy narzędzie zawiesza się w trakcie sesji projektowej, lub uszkadza model, który nie ma kopii zapasowej. Istnieje wiele wiodących narzędzi, które spowodowały utratę wielu godzin pracy z powodu awarii lub uszkodzenia plików.

Współcześnie odporność często można znaleźć w aplikacjach zaimplementowanych w Javie (ochrona czasu pracy JVM) lub w projektach open-source (w całym Internecie, równoległe debugowanie). Najszybszym sposobem sprawdzenia, czy konkretne narzędzie UML jest niezawodne, jest zapytanie w grupie dyskusyjnej, takiej jak comp.object. Na pewno wiele się dowiesz!

Inna strategia, którą należy zastosować, polega na tym, aby narzędzie UML automatycznie zapisywało model w tle w okresowych odstępach czasu.

Platforma

Aby zmaksymalizować inwestycję w narzędzie do modelowania, należy dokładnie rozważyć platformy, na których będzie uruchamiane narzędzie. Czy potrzebujesz oprogramowanie dla systemu Windows lub Unix czy obu? Na jakiej platformie będziesz się rozwijać?

Ostatnio mogłoby wydawać się, że pierwszorzędne, międzyplatformowe graficzne interfejsy użytkownika albo nie będą działać, albo wyglądają za każdym razem inaczej. Przez długi czas było to prawdą (z wyjątkiem aplikacji opartych na HTML) aż do niedawnego pojawienia się interfejsu użytkownika Swinga w języku Java. Jednak narzędzia wieloplatformowe muszą być obsługiwane na platformach towarowych, takich jak Linux, w celu osiągnięcia akceptacji na dużą skalę przez programistów.

Firma Sun początkowo niewiele zrobiła, aby promować Javę w systemie Linux. Jednak ostatnie inicjatywy branżowe, głównie z IBM, które obiecały szerokie wsparcie dla Linuksa na wszystkich platformach sprzętowych i wspierają projekt Apache / Jakarta, teraz szybko przesuwają Javę na Linuksa. Być może dlatego, że IBM przeniósł się do dystrybucji swojej wersji JDK 1.1.8 do głównych dostawców Linuksa, Sun został zmuszony do wspierania dystrybucji w pełni funkcjonalnego JDK 1.2 (Java 2, z Swingiem) dla Linuksa.

Nowe wydania

Chcesz wybrać narzędzie do modelowania, które będzie nadal aktywnie ulepszane za pomocą poprawek, ulepszeń wydajności i dodawania nowych funkcji. W końcu inwestujesz w czas i pieniądze i nie jest łatwo przejść na inne narzędzie do modelowania.

Uważaj na produkty, które zostały nabyte przez większe firmy. Często pierwotni deweloperzy już dawno opuścili firmę, aby przejść do kolejnej wielkiej okazji. Bardzo trudno jest znaleźć utalentowanych programistów, którzy mogą nauczyć się skomplikowanego oprogramowania i chcą go zachować, jeśli nie napisali oryginalnego kodu. Ten scenariusz może się również zdarzyć w przypadku projektów o otwartym kodzie źródłowym ...

Jak ustalić, czy produkt ewoluuje? Zapytaj przedstawiciela handlowego o szczegółowy harmonogram ostatnich wydań i plan działania na przyszłość produktu. Przyjrzyj się stawce, w której wprowadzono funkcje i ulepszenia. Kiedy produkt ma planować obsługę UML 1.3? Czy GUI obsługuje najnowsze konwencje stylu? Możesz również zajrzeć na stronę firmy: czy ogłoszenia o produktach i opinie z zewnątrz są stare, podejrzone.

Lucidchart

HTML5-based. With UML support and real-time collaboration capabilities. It can import Visio files, which is definitely a plus considering that a large userbase for these tools seems to come from more desktop-like modeling environments like Microsoft Visio. Beyond UML, Lucidchart also comes with drawing templates for ER, process modeling, network diagrams and more. They claim to have over 8M users. Impressive! And they just raised over 36M in funding so you can expect a lot of new features and exciting things from Lucidchart. Give it a try.



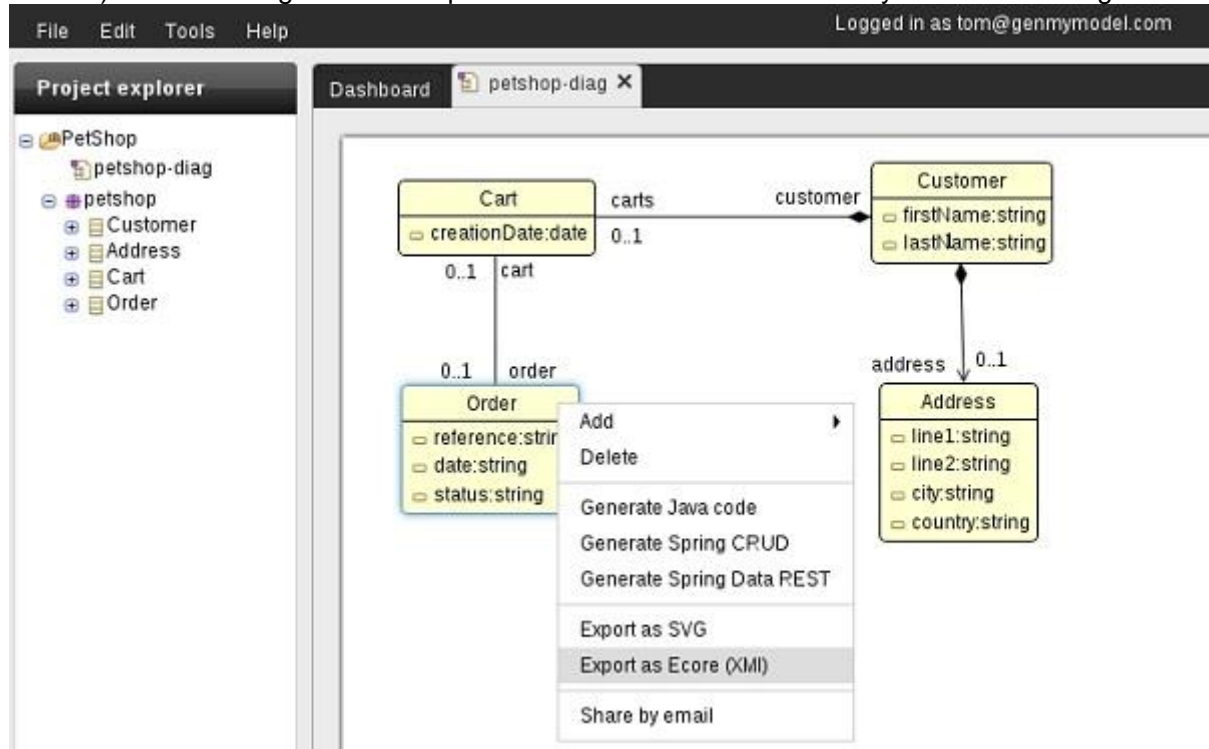
Our flowchart maker works the way you do

GenMyModel

GenMyModel started as a UML only modeling tool but since then it has now expanded to cover also business modeling with Archimate and BPMN support. It offers a centralized model repository for teams (with access right management) enabling easy and simultaneously model collaboration.

Contrary to other tools in this list, this is more a modeling tool than a drawing tool (with its pros and cons, depending on how do you plan to use your models). As such, a key distinction of GenMyModel is its support for exporting models as XML (the "standard" format for exchanging

models) and its code-generation capabilities. Read more about GenMyModel in this blog here.



Gliffy

With a strong emphasis on the collaborative and revision control aspects, Gliffy claims to be “the most widely used online diagramming application” (I wonder how they know that!). Gliffy supports all UML diagrams together with a variety of other kinds of diagrams, including a strong support for BPMN process models.

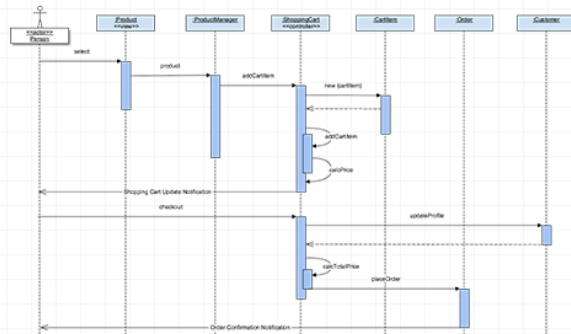
It comes with plugins for Confluence and Jira so clearly it integrates very well with these tools. That’s important if your team uses them and wants to integrate your models in the rest of your discussion /development process.

[PRODUCTS](#) ▾[EXAMPLES](#) ▾[PRICING](#)[SUPPORT](#)[MORE](#) ▾[LOG IN](#)

Make Diagramming a Team Sport

Collaborate on flowcharts, org charts, UML, network diagrams & more.

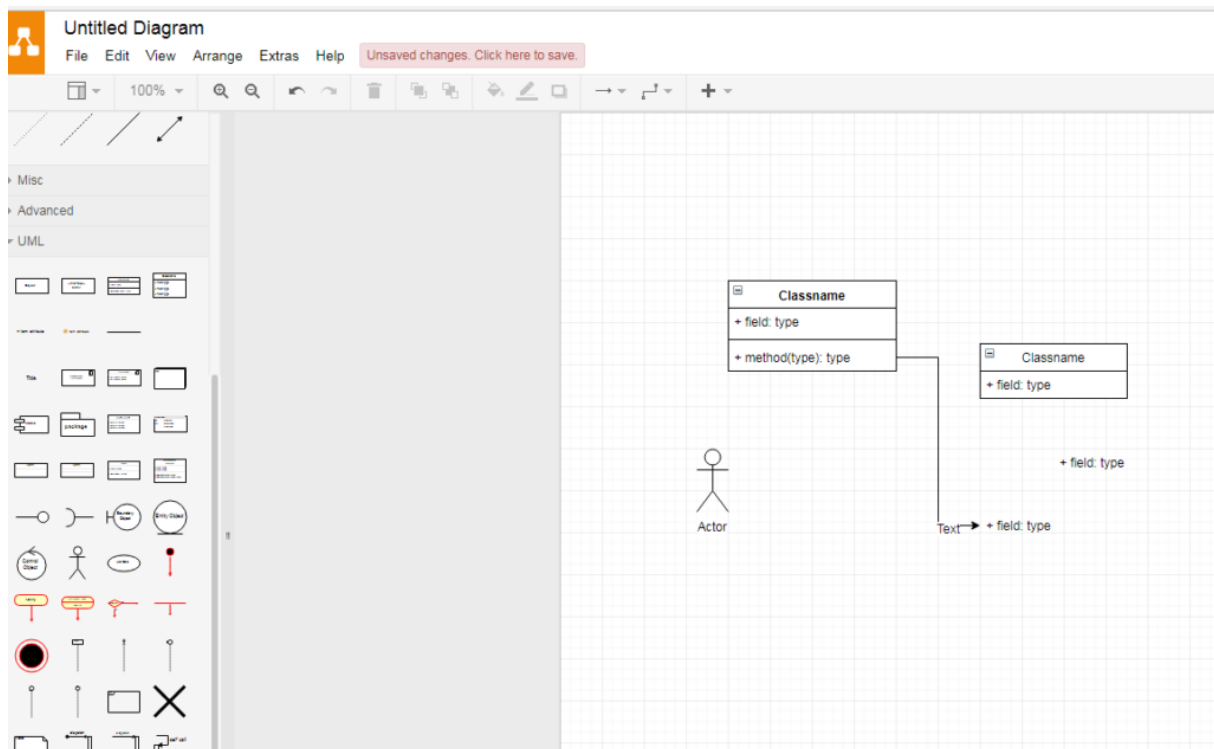
By clicking Start Free Trial, I agree to Gliffy's [Terms of Service](#) and [Privacy Policy](#).

[START FREE TRIAL](#)

Draw.io

Love its simplicity. Click draw.io in your browser and you immediately get an empty canvas to start drawing. It comes with shapes for basic UML, ER and BPMN modeling. Still, it's a clear example of a tool that doesn't really understand at all the semantics of what you're drawing so you can basically do whatever you want and build some very bizarre diagrams. It also fails in the collaborative aspect but it integrates well with Google Drive, Dropbox, OneDrive and others to automatically save the models in your preferred location.

Draw.io is open source and it has been built using the mxGraph library.



Creately


Real-time diagram collaboration. With over 50 diagram types and thousands of examples to get you started. You can also work offline and synchronize your work afterward. The only BUT I see is that, as of today, it requires Flash installed to work. I hope they revisit this issue soon since Adobe will be killing Flash. Sure, not right now, but who wants to use a tool using a doomed technology?

For more about the tool and a short description of its key points see here.



Cacoo

Focus on all kinds of cloud-based diagrams. Really on the “drawing-end” (and not the modeling one). Still, it has some interesting features like the possibility of storing multiple versions of your diagram to track changes and the ability to post comments to give feedback directly on the diagrams. Cacoo supports use case, sequence, class, activity, and state machine diagrams.

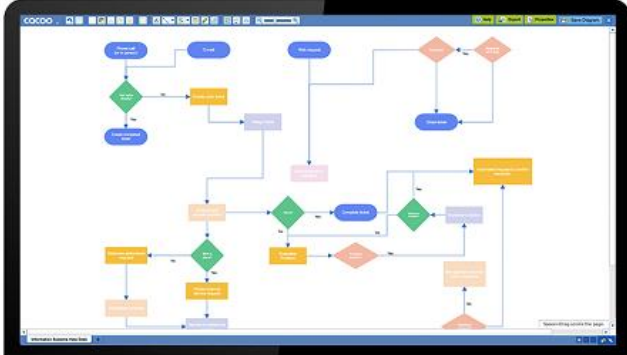
 EN ▾

[Features](#) [Pricing](#) [Templates](#) [For Teams](#) [Try It Free](#) [Log in](#)

Cloud-based diagrams, the easy way.

Diagram maker for companies, teams, and you

[Try It Free](#)



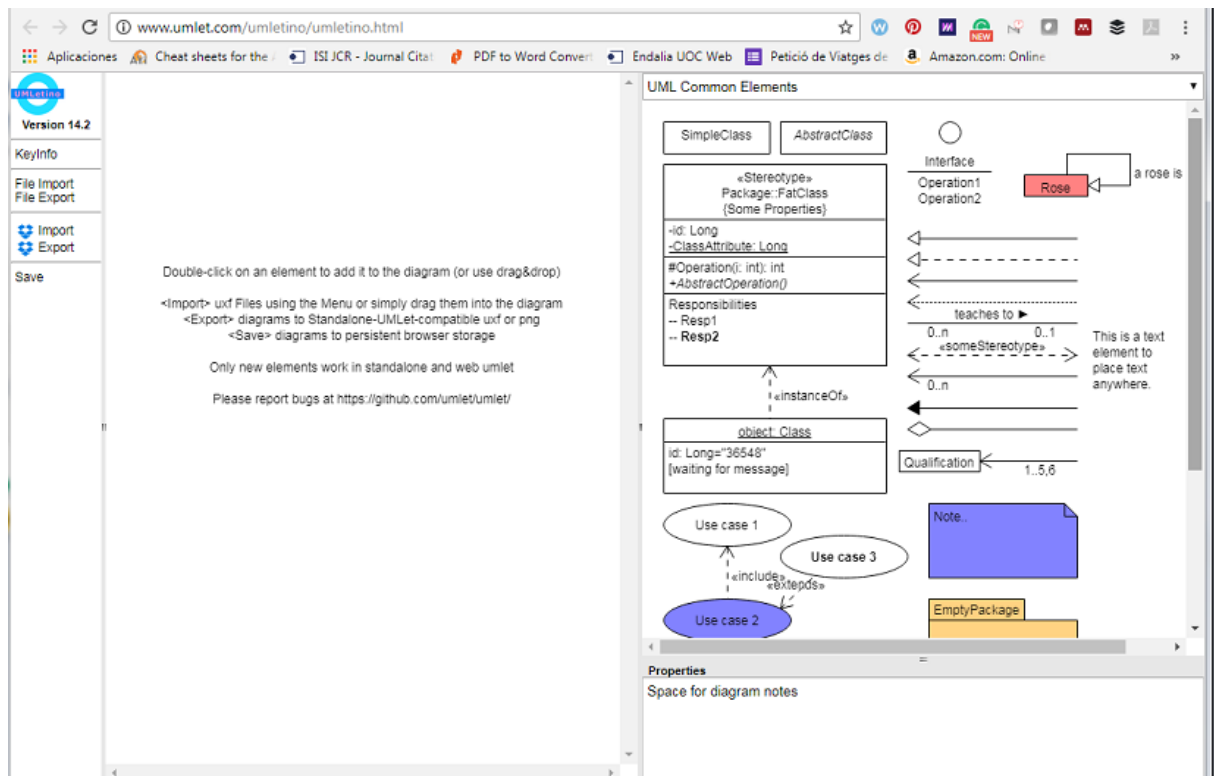
Extensive collection of templates and shapes



UMLetino

A simple but free online UML tool for fast UML diagrams. It runs in your browser, and it does not require any installs. UMLetino is based on UMLet (which is available as a stand-alone tool or Eclipse plugin). Diagrams can be exported as XML or image files.

Despite its limitations (the modeling environment is clearly sub-par when compared with the graphical quality and powerful interface of some of the other tools in this list), if you're looking for a free and easy-to-use UML online tool, this is definitely an option to consider.



Diagramo

A free and open source pure HTML5 Online Flowchart Software (no Flash, Java or other plugins required). Licensed under GPL (nice!). Easy to download and install on your own server if you prefer. The only “but” for the typical readers of this portal is that as a flowchart-focused software, it only covers one kind of UML diagrams: state machines (but there is an easy solution for this

just fork it on Github and extend it yourself
Diagrams can only be saved as images.

). Synchronous collaboration is not possible.



Diagramo is an online flowchart software

- ✓ It's FREE and Open Source ([GPL](#))
- ✓ It's pure [HTML5](#)
- ✓ Share and collaborate
- ✓ Export to SVG, Gif and JPEG
- ✓ Life time support & storage

Try it Now!

No account required.

Download it

Completely free



HTML5
Based ONLY on HTML5. No
Flash, no Java or other
plugins.



**Share and
collaborate**
Share diagrams with
other teammates. Or
make them public if
you want.



Install it on your server
[Download it](#) and deploy it on
your server. All you need is
PHP and a MySQL database.



Customize it!
[Take the sources](#) and
tune them to your needs.



Free! GPL!
Yes, it's a GPL application.
You can follow with its code.



Export to SVG, Gif and JPEG
You can export diagrams in
SVG, Gif or JPEG format.

jsUML2 Editor

The jsUML2 editor is a lightweight UML 2 diagramming tool developed on top of the jsUML2 library, a HTML5/javascript library for UML2 diagramming. Its main objective is to provide web developers an easy way to visualize and edit UML models in their own websites, without other external dependencies and fully executable on the client-side via the browser, unlike other tools in this list.

The jsUML2 library provides an API to allow web developers to include editable diagrams in their own websites making use of the HTML5 technologies supported by the newest web browsers, as well as by current mobile devices. Read more about the library itself [here](#).

Unfortunately, while the editor is still available and you can use it for free, the author recently announced that it's closing the project for the moment due to lack of personal time to work on it.

