**Can the fish in Ed´s aquarium bark?**

No, that was Amber: <http://4.bp.blogspot.com/_H4Kr0LHNMcQ/R2usd9VLG5I/AAAAAAAAAU4/PQJkXFVHHlQ/s1600-h/AmberPerplexed.JPG>

**I played around with EMF binary serialization, it really is compact & fast. So is this just BinaryResourceFactoryImpl then? Is the binary form platform independent?**

Yes, BinaryResourceFactoryImpl. It words for Java on any platform.

**Can we change the underlying data base to use something like Objectivity instead of relational DB?**

Yes, Simon McDuff (CDO team member) is working on exactly that. Basically we are waiting for some IP clearance stuff to make it (the CDO storage adapter) open source. That's the beauty that the persistence vehicle is complete orthogonal. So you can put any type of backend behind it.

**How is CDO doing the object-relational mapping? Is there JPA or JDO underneath or homebrewn?**

The backend integration is completely decoupled from the repository framework. The default integration is our proprietary DBStore. Together with Martin Taal from the Teneo project we developed a Teneo/Hibernate integration. I heard that Teneo supports JPA, so yes, a CDO repository can work with JPA. For object-oriented backends you need no mapping at all. See the preceding question about Objectivity. And by implementing a handful of interfaces you can plug in your own backend integration.

**So the CDO-Teneo integration is not available yet ''off the shelf''?**

It is! We are currently working together with a possible new committer to augment the CDO/Teneo/Hibernate integration team and ensure that this component complies to the same high qualitiy standards as the rest of the CDO framework. Currently the CDO HibernateStore is integral part of the CDO SDK but we are considering to factor it out into separate downloads due to legal issues with the Hibernate dependencies.

**Any advice for a mature open source ODB?**

Due to lack of own experience, no. But we have <https://bugs.eclipse.org/252255> to discuss this topic. Please note that we marked it P5 and HELPWANTED to indicate that we will not be able to address these kinds of integrations ourselves. A perfect opportunity to gain committer privileges !

**Can I have multiple communication plugins available, i.e. AMQP?**

Eike is working to separate out Net4J to better support other communication mechanisms. It would be nice to closely work together with extenders of these new APIs to ensure that they meet concrete expectations. Apart from this effort it should also be possible to integrate new transport technologies via Net4j. Fo this purpose Net4j has a strong separation of the physical transport layer and the application protocl layer (here: CDO protocol).

**Is it possible to tunnel through HTTP, so HTTP on port 80 can go through firewalls for communication?**

Yes, Net4j comes with an (experimental) HTTP transport layer implementation that supports asynchronous bidirectional communication through http connections.

**Am I correct in understanding that a ViewSet is mainly useful when connecting to multiple different repositories?**

Yes, we introduced the explicit notion of a CDOViewSet when we supported external references in the framework, i.e. references from one repository to another repository or completely different resource types. Each EMF RescourceSet always has an associated CDOViewSet (when used with CDO) but in the standard case this ViewSet will contain only a single CDO view or transaction.

**Does each change increment the version of an object?**

No. The version of an object is incremented at the end of a successful commit operation on a the underlying CDOTransaction or CDOXATransaction. In fact the version is part of the primary key of a CDORevision, so a new version is assigned with each new revision of an object.

**When does the view acquire the latest version of an object? The first time you access the object, or every time you access the object?**

Basically the first time you access an object. Sometimes earlier (configurable pre-fetching) and sometimes later (after remote invalidation).

**What happens in case of a conflict (two transactions changing the same object)?**

The most important reaction is that the model repository reliantly rejects the second commit attempt. Depending on the local configuration (session.options.setPassiveUpdateEnabled) you might receive early notification about pending commit failure (conflict state of some objects). All this is about detecting conflicts. Quite new is a pluggable mechanism for custom conflict resolvers. As an example we offer a resolver that reduces the conflict granularity down from object to structural features of an object. See transaction.options.addConflictResolver.

**Can I access the same repository through different sessions too?**

Yes, you can open arbitrary number of sessions to a single repository, even from a single client. But keep in mind that each session maintains a revision cache and should be considered somewhat expensive. Each session can have arbitrary numbers of views / transactions. The objects (empty data wrappers) provided by these views all share the revision instances of the session cache.

**Did I understand Eike correctly that changing an object requires a Transaction, which is the only read-write view?**

Yes.

**Is Audit a ''mode'' of a repository or a view on the repository?**

Both. A CDOAudit is a special CDOView on the client side that enables to look at consistent historical states of the whole object graph. As such it is a client side concept. For this to be possible the repository must be configured to support it. As such it is a mode of the repository. Third, the chosen backend (plus integration) must support this repository configuration. As such it is a capability of the storage implementation.

**If I disable the audit mode, then no revision data is stored anymore, right?**

Yes, if auditing is disabled in the repository, then no historical revisions are stored. Only current revisions. Please note that each storage implementation can have different capabilities with respect to switching auditing on or off. New: the DBStore can switch it off now!

**So when audit is disabled in the server configuration and request an audit object, then what do I get? null or a view on the current revision (the only one available)?**

If the repository is not configured to support auditing you will get an exception when you try to open an audit view on the client side. So you’ll never be able to hold references to old revisions.

**Can I have custom views on the client editor based on different packages (models) I have in the repo? I assume there is a generic editor to handle all the different models I can have in the repo.  I need to have different views on the editor based on different models.**

Yes and yes. CDO supports multiple packages and multiple resources both in the client and in the repository. You can mix them as you want to. And the generic CDOEditor supports all of this, too.

**Do you have references of production apps based on CDO?**

Yes, we have. Although I am often surprised to learn that many companies use CDO for a long time before I even get aware of it through some newsgroup questions. I am planning to open a public poll to gather more feedback, stay tuned. We know of many applications in the modeling industry (tooling) as well as in space and aeronautics, automotive, defense, asset management, and so on. If you know of any applications please tell us about!

**Why is the server config file not an EMF model?**

The main reson is that formerly we had no other EMF requirement on the server side of CDO. This will change in the near future to better support custom EdataTypes and Eoperations in server-side queries. Together with our new Definition Framework (<https://bugs.eclipse.org/246623>) this will make a perfect fit to configure a CDO server.

**Any parallels/differences to MS Oslo Repository (i.e. in philosophy/objectives)?**

Ed Merks talked about Oslo in this blog: <http://ed-merks.blogspot.com/2008/11/modeling-pilgrimage-to-france.html> . The idea of CDO is to support any repository, not just one specific one. I think that can make some aspects of Oslo much slicker.

**Does the CDO user interface use standard properties viewer? So the Databinding frameworks can sit on top to create custom UIs and Views?**

Yes, CDO is pretty much invisible for the rest of the application. Standards should always be usable.

**What is the state of CDO Browser project?**

We've recently added Common Navigator and Team Provider support. There is also EFS integration on the way (contribution). It is the main focus of the UI sub component to deliver high-quality and re-usable user interface components that make perfect use of the scalable core components of CDO.

**In the demo, when you change something in one client, and it appears on the other client, what's the API like for handling that? Do you have to use listeners? Or does it just look like the model was changed by another thread?**

Both, you receive EMF Notifications (depending on the local configuration with or without change deltas) and you also can listen to some special events sent by the server. See <http://wiki.eclipse.org/FAQ_for_CDO_and_Net4j#How_can_I_react_to_remote_changes_to_my_objects.3F> for details.

**Wow! Team Provider, cool! It would be kind of natural to have something like the compare view for conflicts. Can team integration be used to handle conflicts?**

Team integration is very new and work in progress. But yes, it is our plan to make all the respective CDO core functionality available through the well-known Team support of the Eclipse DIE, including compare and merge. Consider also that while CVS has file level granularity, CDO can handle two clients updating different properties of an individual object. You can't get more fine grained than this...

**I saw that a CDOResource can have a „resource path“. Does that mean the "/folder/company" string is processed and treated as an IPath or something anagolous?**

A special obejct hierarchy is defined for that, CDOResourceNode (abstract), CDOResourceFolder and CDOResource. They form kind of a virtual file system for resources within a model repository. They can be navigated and queried just like normal objects (in fact they are EObjects). Each such CDOResourceNode has a unique path, but it is always presented as a String. You can convert it easily to an IPath.

**Does CDO provide a Validation mechanism?**

At the CDO client side you can apply any validation mechanism that you can get to work with regular EMF models. CDO does not offer own solutions in this area. On the server side you can hook your own vetoable ReadHandlers and WriteHandlers with a particular repository. In the future you will also be able to use EMF based technologies like EMF Query, Validation and OCL on the server side, together with your own validators written in Java.

**Are there query facilities for querying the models in the repo?**

We have a very flexible query framework that can support any type of query language and deliver query results synchronously as well as asynchronously from the server to the client. You can hook either backend-specific or backend-agnostic language handlers into a repository. We would appreciate the contribution of such query handlers. And we are working on the integration of a common query language (OCL).

**Any thoughts on data migration? Data migration and data import from/to XMI?**

Import and export to any other persistence style that is supported by EMF is also supported by CDO. We have examples for XML/XMI import and export. Automatic data migration, as needed after model evolution, is currently not supported, although we started discussion on this topic in https://bugs.eclipse.org/256856

**How about locking, one client can lock the object so that the other clients not allow to change till it's unlocked? Is there any tree-locking facilities in CDO and/or EMF?**

Yes and no. We have explicit locking on object level. You can acquire read and qrite locks one one or several objects at once. The contained sub tree of objects is not included! Locks are automatically released when the view/transaction is closed, optionally at commit time. This locking support is part of the CDO core.

**Would it be possible to run the CDO server code within an app server - would that make sense at all?**

Yes and yes. Each single component of CDO can run in OSGi, standalone or in any imaginable container. Embedding the repository into an JEE server makes particular sense, when the CDO client is a web application in the same container. Then you can use Net4js JVM connector to bypass socket transport. In the near future we will come up with even more direct connection methods (i.e. without Net4j in between).

**Though it's called connected data objects, it there some way of using CDO in a kind-of offline mode?**

Currently not. When the connection is terminated or breaks down the CDO session is automatically closed, along with all open views and transactions. But due to the tremendous demnad for and usefulness of an offline mode we have made this feature our current top priority. The problem itself is two-fold. Since demand loading will not work when offline we need to checkout partial copies of the central object graph. The ability to cache revisions in the local file system will also provide for even better scalability because will use this ability to page out dirty parts of local transactions. For the switch back to online mode we need a sync/merge mechanism for objects. This has recently been added to support the pluggable conflict resolvers of transactions.

**Speaking of transformation, QVT/ATL over EMF/CDO should be possible, right? Would it perform well?**

You should be able to use CDO backed models just like any other "normal" EMF model. CDO is effectively transparent so pretty much every normal EMF thing applies. Regarding performance, there is a lot that you can configure in CDO. See <http://wiki.eclipse.org/Tweaking_CDO_Performance> for details.

**Which newsgroups to submit thoughts to?**

Use the EMF newsgroup and prefix with [CDO]: See http://wiki.eclipse.org/CDO\_Project\_Resources#Support\_and\_Feedback

**We need to save locks when user is offline, is that possible?**

Currently, when a user is offline, the session is terminated (offline mode for a session is work in progress). Maintaining locks beyond the end of a session is currently not supported by CDO. Locks are automatically released when the view/transaction is closed, optionally at commit time.

**Is there dynamic CDO?**

Yes. You can register dynamic EPackages with a CDOSession and commit them to the repository. Note, that you can generally only register top-level packages with a CDO session. Nested packages are registered implicitely if they have a namespace URI.

**Do I have to regenerate my models?**

Currently yes, or you must use a dynamic form of your model. We are working together with EMF Core on a mechanism to support „legacy models“ (i.e. those that are generated, but not for CDO) in CDO. See <https://bugs.eclipse.org/247226> for details. Please note that regeneration does not require the Ecore model to be changed! But the regeneration for CDO removes all member fields from the generated classes (reflective delegation) and your hand-written modifications that use these member fields (they should use their accessors!) would not compile anymore.

**What about the new EMF index project and server side queries?**

EMF Index will work on the EMF side not directly on the servery/repository side. But we are working closely with them to ensure that CDO can nicely fit into an index provider framework.

**Are the CDO webinar and the webinar slides available online?**

Yes, Lynn will make both available through Eclipse Live: <http://live.eclipse.org/node/659> . And I’ll check the slides into the CVS and make them available through the CDO wiki: <http://wiki.eclipse.org/CDO>

**Is there kind of transaction Rollback mechanism?**

CDOTransaction and CDOXATransaction have rollback() methods.We even have save points which you can roll back to. There's a save point snippet in the webinar slides.

**If it was just about storing EMF objects in an RDBMS, Teneo would be sufficient I'd think. So I was wondering what other requirements you could have for the whole "bus" thing?**

For instance a distributed shared model or the client side scalability for really huge models. CDO provides a clever memory usage with garbage collector support and stuff like partial object loading, partial collection loading. In this sense CDO is like the virtual memory of EMF (Steve suggested: „Demand Paged Virtual Modelling!“).