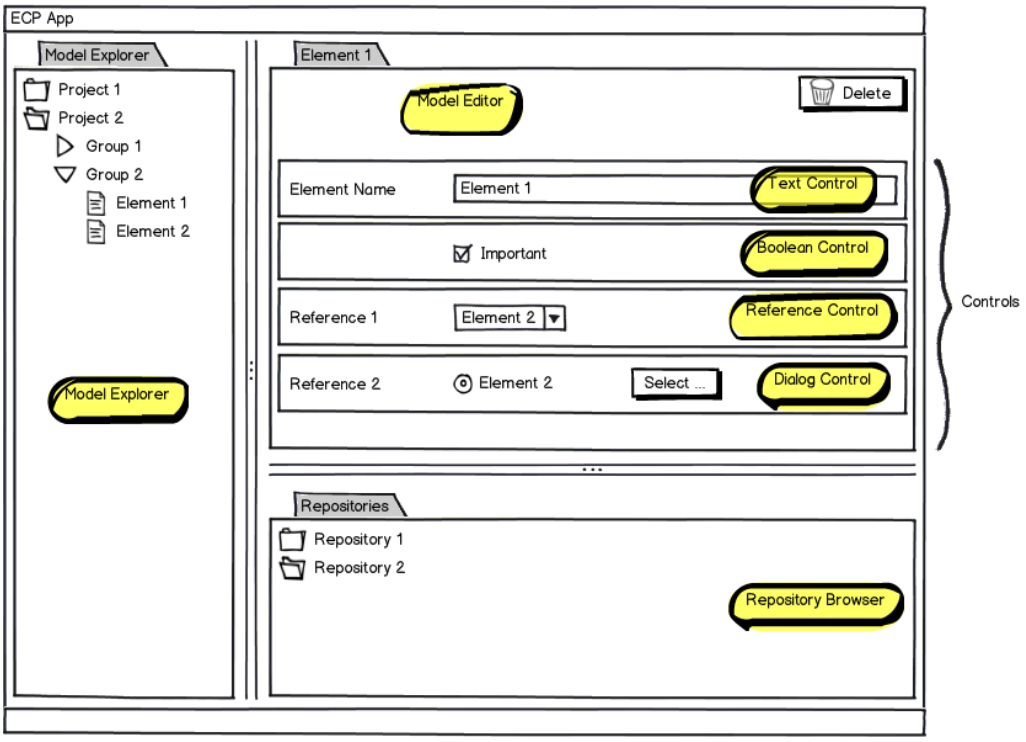
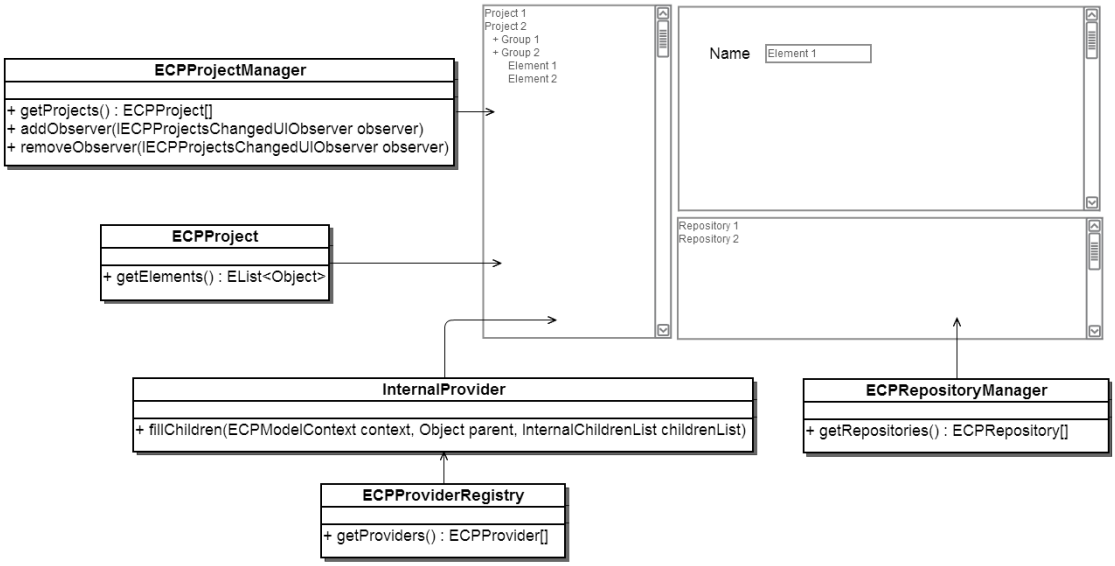
ECP App Proposal





Custom Annotations for ECP:

- Feature / Category order: org.eclipse.emf.ecp.internal.editor.descriptor.AbstractAttributeDescriptor<T>

TODOs:

- draw overview chart that shows the workings (esp. hidden stuff like Provider, ModelElementOpener etc.)

General Questions

Is the analogy correct?

ECP Element	EMF Element
Repository	ResourceSet
Project	Resource

If so why can the Workspace not be modeled that way? Notificatons?

Answer: Kind of. The Workspace implementation acutally uses this approach. Other implementations don't. Reasons are lazy loading, etc. (e.g. CDO)

Repository Browser

The Repository Browser provides the following functionality

- add repositories
- delete repsositories
- edit repositories
- connect to repositories

The central service to access the repositories is the ECPRepositoryManager:

```
public interface ECPRepositoryManager {
 ECPRepositoryManager INSTANCE =
org.eclipse.emf.ecp.internal.core.ECPRepositoryManagerImpl.INSTANCE;
 ECPRepository getRepository(Object adaptable);
 ECPRepository getRepository(String name);
 ECPRepository[] getRepositories(); (should be List<?> ?!)
 ECPRepository addRepository(ECPProvider provider, String name, String label,
String description,
  ECPProperties properties);
}
```

The ECPProviders are registered with the ECPProviderRegistry:

```
public interface ECPProviderRegistry {
 ECPProviderRegistry INSTANCE =
org.eclipse.emf.ecp.internal.core.ECPProviderRegistryImpl.INSTANCE;
 ECPProvider getProvider(Object adaptable);
 ECPProvider getProvider(String name);
 ECPProvider[] getProviders();
 void addProvider(ECPProvider provider);
 void removeProvider(String name);
 void addObserver(IECPProvidersChangedObserver changeObserver);
 void removeObserver(IECPProvidersChangedObserver changeObserver);
}
```

ECPRepositoryManager INSTANCE

- should be removed

How can I add a repository?
How can I delete a repository?
How can I list the repositories?
State Updates?
Minimal Set of Supported Operations:

- Add a repository (specific dialog?)

- Delete a repository (specific dialog?)

ECProvider: e.g. WorkspaceProvider (does not require config) needs to be registered
Where do I get the Providers from?

- Pointer: DefaultUIProvider finds the data providers; ECPProviderRegistry

Model Explorer

The Model Explorer views the projects provided by the open repositories and their contents. The entry point to the projects is the ECPProjectManager:

```
public interface ECPProjectManager {
 ECPProjectManager INSTANCE =
org.eclipse.emf.ecp.internal.core.ECPProjectManagerImpl.INSTANCE;
 ECPProject createProject(ECPProvider provider, String name) throws
ProjectWithNameExistsException;
 ECPProject createProject(ECPProvider provider, String name, ECPProperties
properties)
  throws ProjectWithNameExistsException;
 ECPProject createProject(ECPRepository repository, String name, ECPProperties
properties)
  throws ProjectWithNameExistsException;
 ECPProject cloneProject(ECPProject project);
 ECPProject getProject(Object adaptable);
 ECPProject getProject(String name);
 ECPProject[] getProjects();
 void addObserver(IECPProjectsChangedUIObserver observer);
 void removeObserver(IECPProjectsChangedUIObserver observer);
}
```

It provides the projects and allows to register change listeneres that are notified when the projects change. The contents of the projects can be retrieved form the ECPProjects:

```
public interface ECPProject extends ECPElement, ECPModelContext,
ECPRepositoryAware, ECPPropertiesAware, ECPCloseable,
 ECPDeletable, IEditingDomainProvider, IAdaptable {
 String TYPE = "Project";
 EList<Object> getElements();
 Collection<EPackage> getUnsupportedEPackages();
 void setVisiblePackages(Set<EPackage> visiblePackages);
 Set<EPackage> getVisiblePackages();
 Set<EClass> getVisibleEClasses();
 void setVisibleEClasses(Set<EClass> visibleEClasses);
 Iterator<EObject> getReferenceCandidates(EObject eObject, EReference eReference);
 void saveProperties();
 void saveModel();
 boolean isModelDirty();
 void delete(Collection<EObject> eObjects);
 boolean isModelRoot(Object object);
 boolean contains(Object object);
}
```

The children of an object in the project can be retrieved using InternalProvider:

```
public interface InternalProvider extends ECPProvider, ECPProviderAware,
ECPModelContextProvider,
 InternalRegistryElement, AdapterProvider {

 public enum LifecycleEvent {
  CREATE, INIT, DISPOSE, REMOVE;
 }

 ComposedAdapterFactory EMF_ADAPTER_FACTORY = new ComposedAdapterFactory(
  ComposedAdapterFactory.Descriptor.Registry.INSTANCE);
 AdapterProvider getUIProvider();
 void setUIProvider(AdapterProvider uiProvider);
 EditingDomain createEditingDomain(InternalProject project);
 boolean isSlow(Object parent);
 void fillChildren(ECPModelContext context, Object parent, InternalChildrenList
childrenList);
 void handleLifecycle(ECPModelContext context, LifecycleEvent event);
 EList<? extends Object> getElements(InternalProject project);
 Collection<EPackage> getUnsupportedEPackages(Collection<EPackage> ePackages,
InternalRepository repository);
 Iterator<EObject> getLinkElements(InternalProject project, EObject modelElement,
EReference eReference);
 void doSave(InternalProject project);
 boolean isDirty(InternalProject project);
 void delete(InternalProject project, Collection<EObject> eObjects);
 void cloneProject(final InternalProject projectToClone, InternalProject
targetProject);
 boolean modelExists(InternalProject project);
 Notifier getRoot(InternalProject project);
 ECPProject[] getOpenProjects();
 boolean contains(InternalProject project, Object object);
}
```

INSTANCE should be removed

How do I list the projects?

How do I add a project?

How do I delete a project?

What's the difference between a Resource and an ECPProject?

How do I open / edit an element contained in the project?

Get Children: DefaultProvider.fillChildren(ECPModelContext, Object, InternalChildrenList) (InternalProvider)

Listeners on the Project!!

All TreeNodes should register with the ECProject for updates (performance / observer bus / treebus)

Model Editor

ECPEditorContext

MEEditorInput

```
public class MEEditorInput implements IEditorInput {
 private EStructuralFeature problemFeature;
 private ECPEditorContext modelElementContext;
 // [...]
}
```

What is the problemFeature used for? (obsolete)

Model Editor Control

ECPControlContext

```
public interface ECPControlContext {
 DataBindingContext getDataBindingContext();
 EditingDomain getEditingDomain();
 void openEditor(EObject o);
 void addModelElement(EObject eObject,EReference eReference);
 EObject getModelElement();
 EObject getNewElementFor(EReference eReference);
 EObject getExistingElementFor(EReference eReference);
 boolean isRunningAsWebApplication();
}
```

DatabindingContext

- Is not an interface
- Works with neither Notifiers nor JavaFX natively

openEditor: can this be handled by the ModelOpener?

addModelElement: can this be done the standard way?

getNew / getExistingElementFor: can this be done by a central service?

isRunningAsWebApplication: is that really necessary?

Can we handle validation via the DatabindingContext?