Eclipse plugin issue

If you generate an Eclipse plugin out of the ecore metamodel and try to run the plugin to load the .iso20022 model, you may have the impression that Eclipse is hanging.

In fact, The Eclipse configuration is not hung; it is simply very slow to open resources with the default EMF configuration (because it looks up eagerly all cross-references in the file, instead of doing a two-pass parsing).
I suggest replacing the default generated XMI resource factory by one that contains the options for performance as advised by the EMF book.

In order to do so, the following extension has to be added to plugin.xml (in the model plugin):

```
   <extension
         point="org.eclipse.emf.ecore.extension_parser">
      <parser
            class="iso20022.util.PerformantXMIResourceFactoryImpl"
            type="iso20022">
      </parser>
   </extension>
```

And the following class has to be created (in this example, in iso20022.util):

```
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import org.eclipse.emf.common.util.URI;
import org.eclipse.emf.ecore.resource.Resource;
import org.eclipse.emf.ecore.resource.impl.ResourceFactoryImpl;
import org.eclipse.emf.ecore.xmi.XMIResource;
import org.eclipse.emf.ecore.xmi.XMLParserPool;
import org.eclipse.emf.ecore.xmi.XMLResource;
import org.eclipse.emf.ecore.xmi.impl.XMIResourceImpl;
import org.eclipse.emf.ecore.xmi.impl.XMLParserPoolImpl;

public class PerformantXMIResourceFactoryImpl extends ResourceFactoryImpl {

       private List<Object> lookupTable = new ArrayList<Object>();

       private XMLParserPool parserPool = new XMLParserPoolImpl();

       protected void configureResource(XMIResource resource){
              Map<Object,Object> saveOptions = resource.getDefaultSaveOptions();
              saveOptions.put(XMLResource.OPTION_CONFIGURATION_CACHE,
Boolean.TRUE);
              saveOptions.put(XMLResource.OPTION_USE_CACHED_LOOKUP_TABLE,
lookupTable);
              saveOptions.put(XMLResource.OPTION_ENCODING,"UTF-8");
//            saveOptions.put(XMLResource.OPTION_USE_FILE_BUFFER, Boolean.TRUE);


              Map<Object,Object> loadOptions = resource.getDefaultLoadOptions();
              loadOptions.put(XMLResource.OPTION_DEFER_ATTACHMENT, Boolean.TRUE);
              loadOptions.put(XMLResource.OPTION_DEFER_IDREF_RESOLUTION,
Boolean.TRUE);
```

```
			loadOptions.put(XMLResource.OPTION_USE_DEPRECATED_METHODS, 
Boolean.FALSE);
			loadOptions.put(XMLResource.OPTION_USE_PARSER_POOL, parserPool);
	}

	@Override
	public Resource createResource(URI uri) {
		XMIResource resource;
		resource = new XMIResourceImpl(uri){
			@Override
			protected boolean useIDs() {
				return false;
			}
		};

		configureResource(resource);
		return resource;
	}
}
```