



Eclipse Modeling Framework. Building DSL IDE support in Eclipse

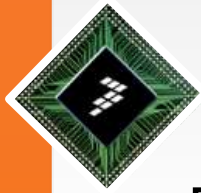
Dorin Ciuca – Software Developer



10-July-2013

Freescale, the Freescale logo, AlliVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, the SafeAssure logo, SMARTMOS, TurboLink, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2012 Freescale Semiconductor, Inc.





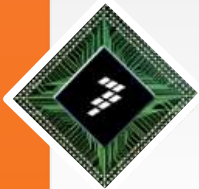
AGENDA

- Eclipse Modeling Framework Overview
 - What is EMF?
 - Ecore – the core of EMF
 - Why using EMF? Difference than a POJO model
 - Eclipse Modeling Ecosystem
- Develop IDE support to your custom domain specific language using Xtext
 - What is a DSL?
 - What is Xtext? How does it work?

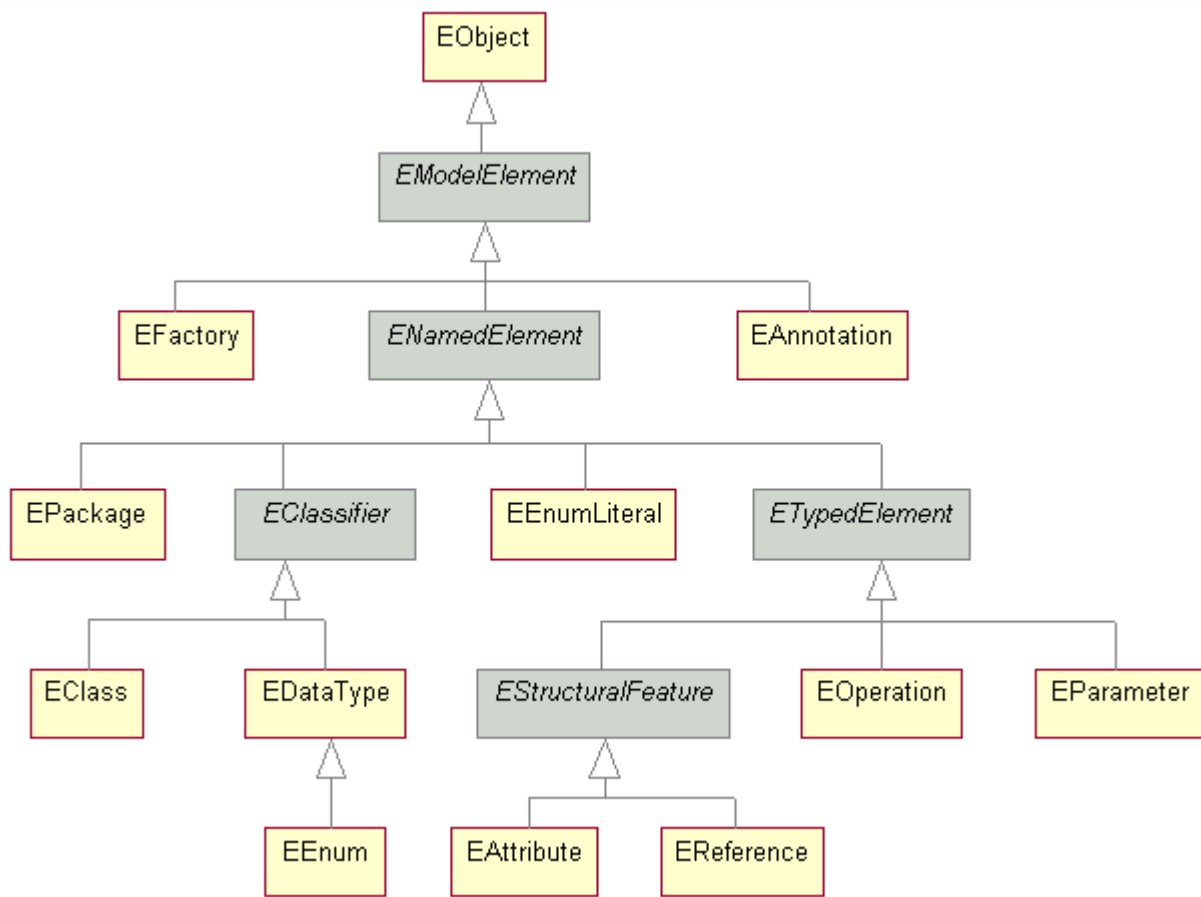


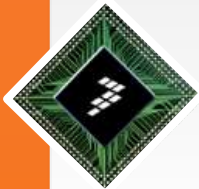
What is EMF?

- <http://www.eclipse.org/emf>
- The EMF project is a modeling framework and code generation facility for building tools and other applications based on a structured data model. From a model specification described in XMI, EMF provides tools and runtime support to produce a set of Java classes for the model, along with a set of adapter classes that enable viewing and command-based editing of the model, and a basic editor.

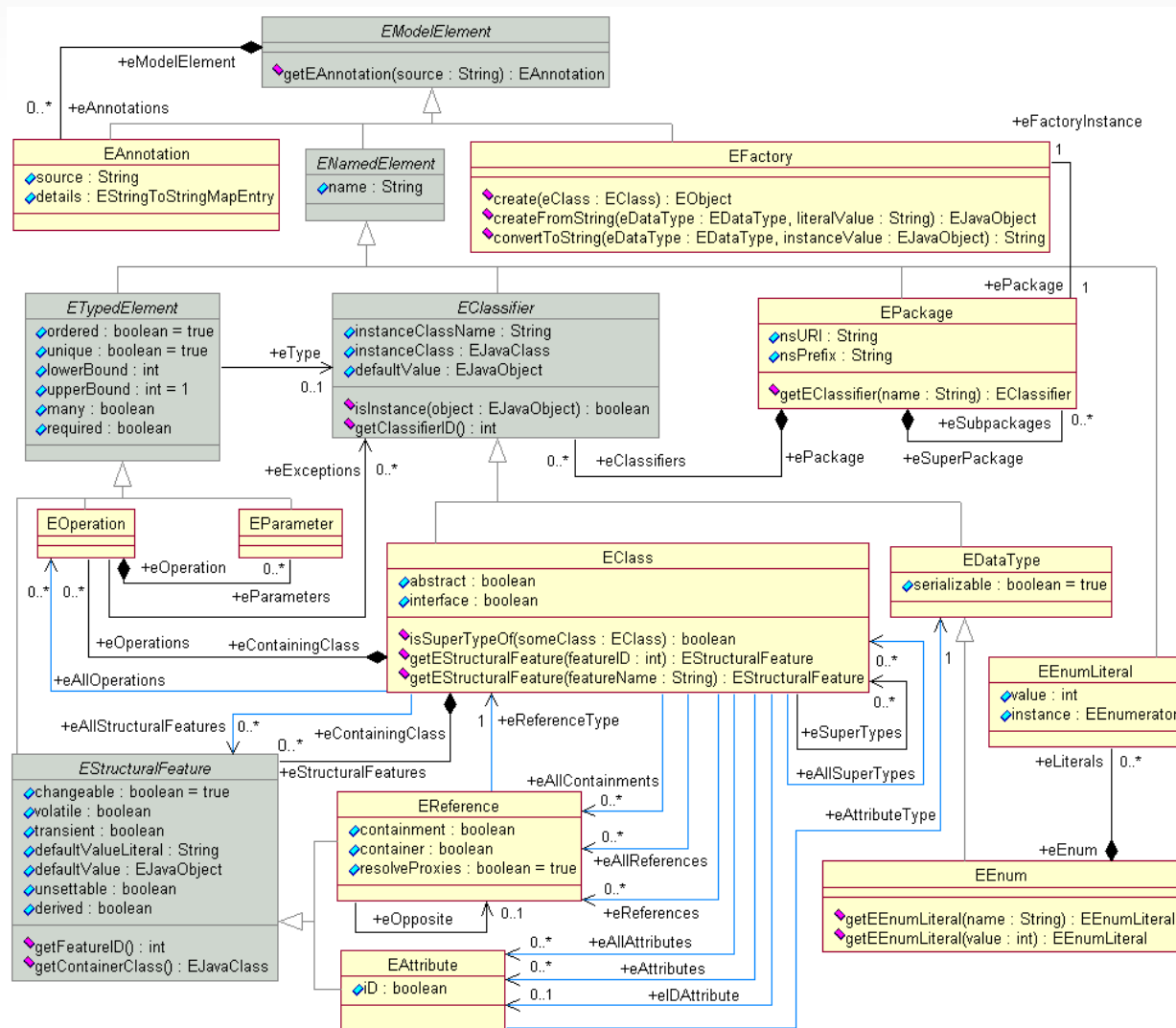


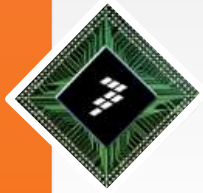
Ecore – the core of EMF





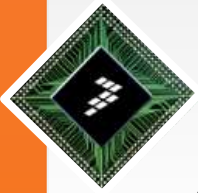
Ecore – the core of EMF





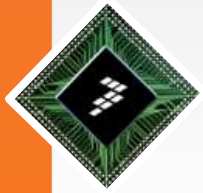
Why using EMF?

- flexibility of the model design
- flexibility of the model interconnection with other components




Eclipse Modeling Ecosystem

- **Abstract Syntax Development**
 - Model Core
 - Model Query (MQ)
 - Model Transaction (MT)
 - Validation Framework (VF)
- **Concrete Syntax Development**
 - Graphical Modeling Project (GMP)
 - Textual Modeling Framework (TMF)
- **Model Development Tools (MDT)**
- **Model Transformation**
- **Technology and Research**



Create EMF Model for Databinding

-  Exercise – Create EMF Model for data binding sample project
- Update previously created data binding example with new EMF model



Building DSL IDE support in Eclipse

Dorin Ciuca – Software Developer



10-July-2013

Freescale, the Freescale logo, AlliVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, the SafeAssure logo, SMARTMOS, TurboLink, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2012 Freescale Semiconductor, Inc.





Ce este un DSL?

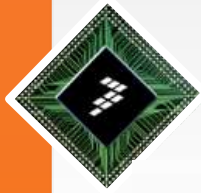
- DSL (*Domain-Specific Language*) reprezinta un limbaj de programare minimal orientat pe un anumit domeniu. Ex: SQL, limbaje XML cum e ANT etc
 - GPL (*General Purpose Language*) limbaj de programare general Ex: Java, C, C++, C# etc
-
- Folosirea unui GPL pentru rezolvarea unei probleme dintr-un anumit domeniu este posibila, dar nu e neaparat cea mai indicata solutie. Un DSL pentru acel domeniu ar trebui sa ajute la: cresterea productivitatii, claritatii codului, mentinere si modificari mai usoare etc.
 - XML poate fi folosit pentru definirea de DSL-uri, dar are ca dezavantaj major sintaxa generalizata care nu e usor de urmarit de catre oameni



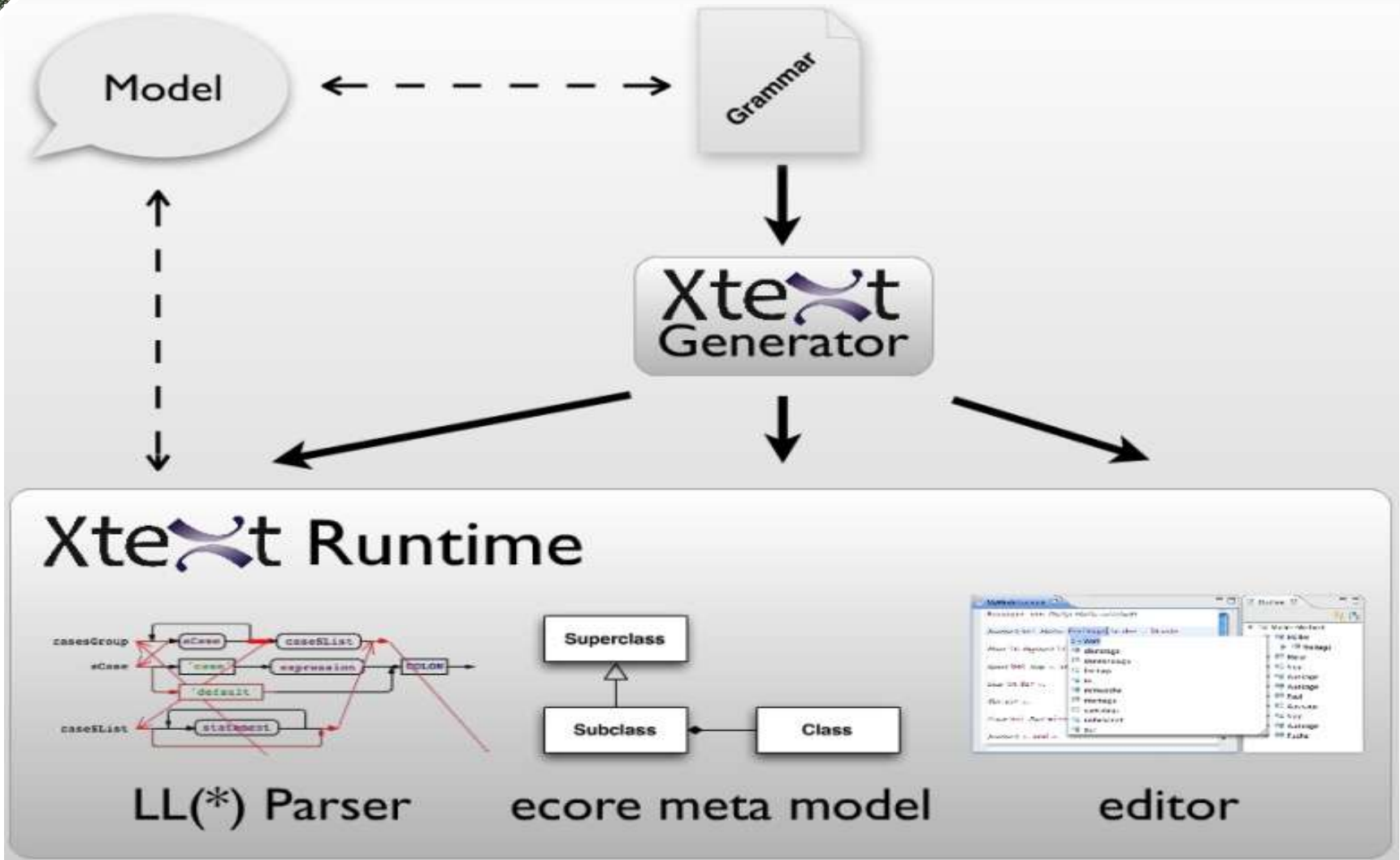
Ce este Xtext?

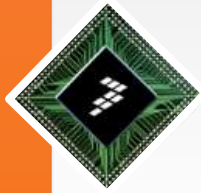


- Conform <http://www.eclipse.org/Xtext> este:
 - un framework pentru dezvoltarea de limbaje conform
 - folosit pentru
 - a crea un limbaj nou
 - a adauga suport pentru un limbaj existent
 - un mic DSL
 - un GPL complex
- Xtext permite adaugarea de IDE support in Eclipse pentru un limbaj, asemanator cu suportul existent deja pentru Java in Eclipse IDE.



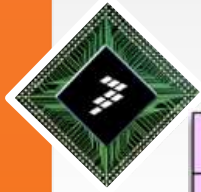
Cum functioneaza?



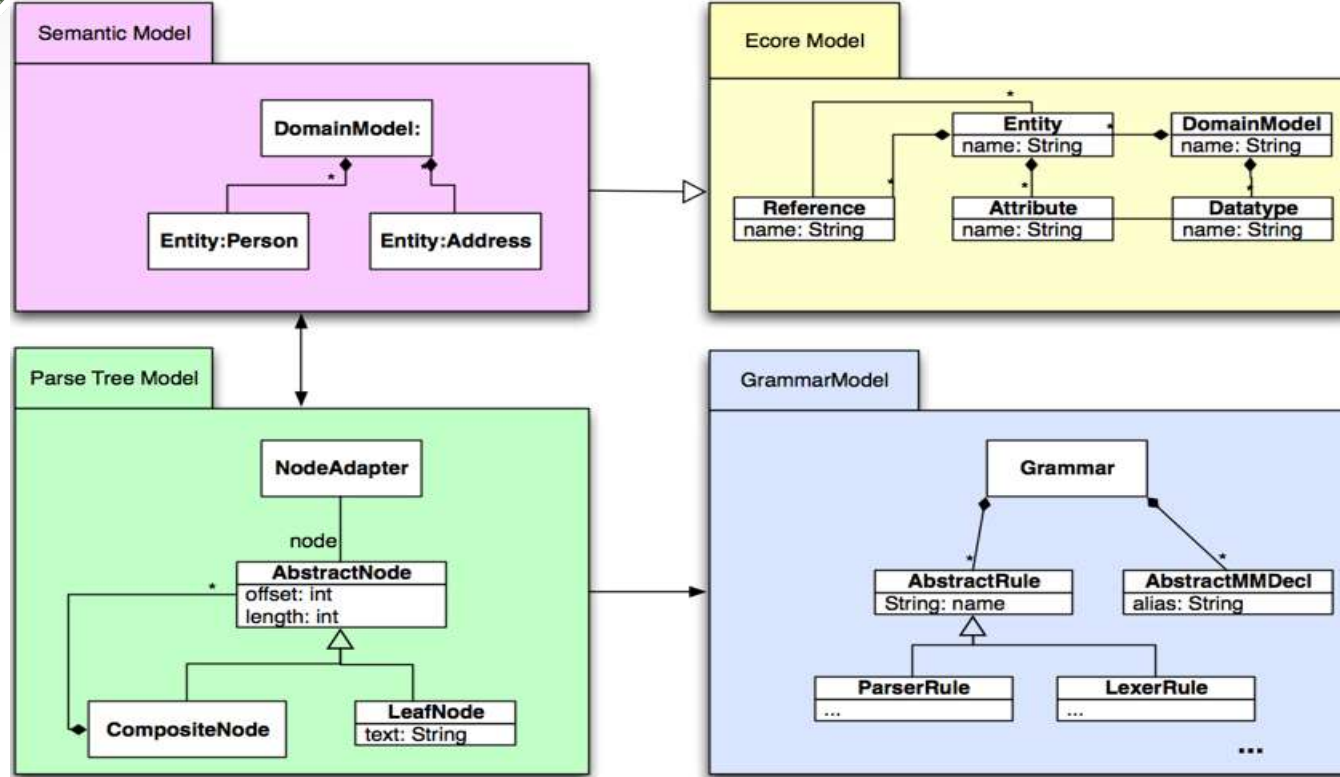


Cum functioneaza? (continuare)

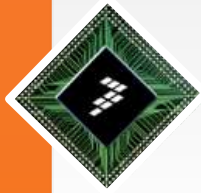
- Porneste de la gramatica limbajului care trebuie definita de user
- Foloseste ANTLR pentru a genera un parser conform gramaticii (<http://www.antlr.org>)
- Genereaza metamodelul pentru abstract syntax tree (AST) bazat pe ecore (de fapt este un graf, dar il numim traditional AST)
- Genereaza un eclipse plug-in care va contine suportul pentru IDE bazat pe AST si ParseTree
- Genereaza optional un eclipse plug-in care contine un generator pentru limbajul definit



Cum functioneaza? (continuare)

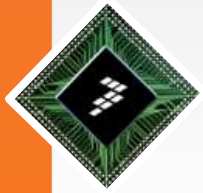


- La runtime, parserul populeaza cele 2 modele AST si ParseTree activand suportul in IDE.



Alte detalii

- AST-ul rezultat fiind un model EMF, se poate usor interconecta cu alte framework-uri bazate pe EMF ca GMF
- Xtext se poate folosi si ca aplicatie java standalone, nu e bazat pe Eclipse OSGI.



Gramatica

- Gramatica este inima Xtext-ului
- Format EBNF
- Reguli
 - terminale
 - de parsare
- Gramatica asigura:
 1. lexing
 2. parsing
 3. linking
 4. validation



Demo – mem files - grammar

```
grammar com.freescale.dsl.mem.MemInitFileDsl with org.eclipse.xtext.common.Terminals
```

```
generate memInitFileDsl "http://www.freescale.com/dsl/mem/MemInitFileDsl"
```

Model:

```
'target' target = ID
'reservedchar' reservedChar = HEX2DIGITS
blocks += Block*;
```

Block :

```
Range | Reserved;
```

Range:

```
'range' loAddr=HEX hiAddr=HEX sizeCode=Size access=Access;
```

Reserved:

```
'reserved' loAddr=HEX hiAddr=HEX;
```

enum Access:

```
ReadWrite | ReadOnly;
```

enum Size:

```
ANY = 'Any' | TWO = '2' | FOUR = '4';
```

```
terminal HEX2DIGITS : '0x' ('0'..'9'|'A'..'F') ('0'..'9'|'A'..'F');
```

```
terminal HEX : '0x' ('0'..'9'|'A'..'F')+;
```

Demo – mem files - editor

The screenshot shows the Eclipse IDE interface with the following components:

- Title Bar:** Java - test/MS8144ADS_Memory_Example.mem - Eclipse SDK
- Package Explorer (Left):** Shows a project named 'test' containing a file 'MSC8144ADS_Memory_Example.mem'.
- Main Editor (Center):** Displays the memory map file with the following content:


```

reserved    0xFF000000  0xFFEFFFDF  // Reserved interr
range       0xFFEFFF00  0xFFEFFFFF  Any ReadWrite // OCE (TODO)
reserved    0xFFFF0000  0xFFFF003F  // Reserved interr
range       0xFFFF0040  0xFFFF005F  4 ReadWrite // EPIC
reserved    0xFFFF0056  0xFFFF0056  // Reserved interr
range       0xFFFF0056  0xFFFF0057  4 ReadWrite // EPIC P_DE
reserved    0xFFFF0056  0xFFFF007F
range       0xFFFF0080  0xFFFF0080  4 ReadWrite // DC_CRO, DC_CR1
range       0xFFFF0080  0xFFF00800  DC_CR2
reserved    0xFFFF0080  // range 0xFFF // DC_LRM, DC_
reserved    0xFFFF0082  // Disabled to
range       0xFFFF0082  DC_DBG_DATA, DC
reserved    0xFFFF0083
range       0xFFFF00C0  IC_CRO, IC_CR1
range       0xFFFF00C0  IC_CR2
reserved    0xFFFF00C0  // range 0xFFF00C00 0xFFF00C01 1 ReadWrite // IC_LRM, IC_
      
```
- Outline (Right):** Shows a tree view of the memory map for 'MSC8144', listing various memory addresses from 0x00000000 to 0xFFFF06000.
- Problems (Bottom):** Shows 0 items.
- Bottom Bar:** Includes tabs for '@ Javadoc' and 'Declaration', and a status bar with 'Writable', 'Insert', and '51 : 22'.