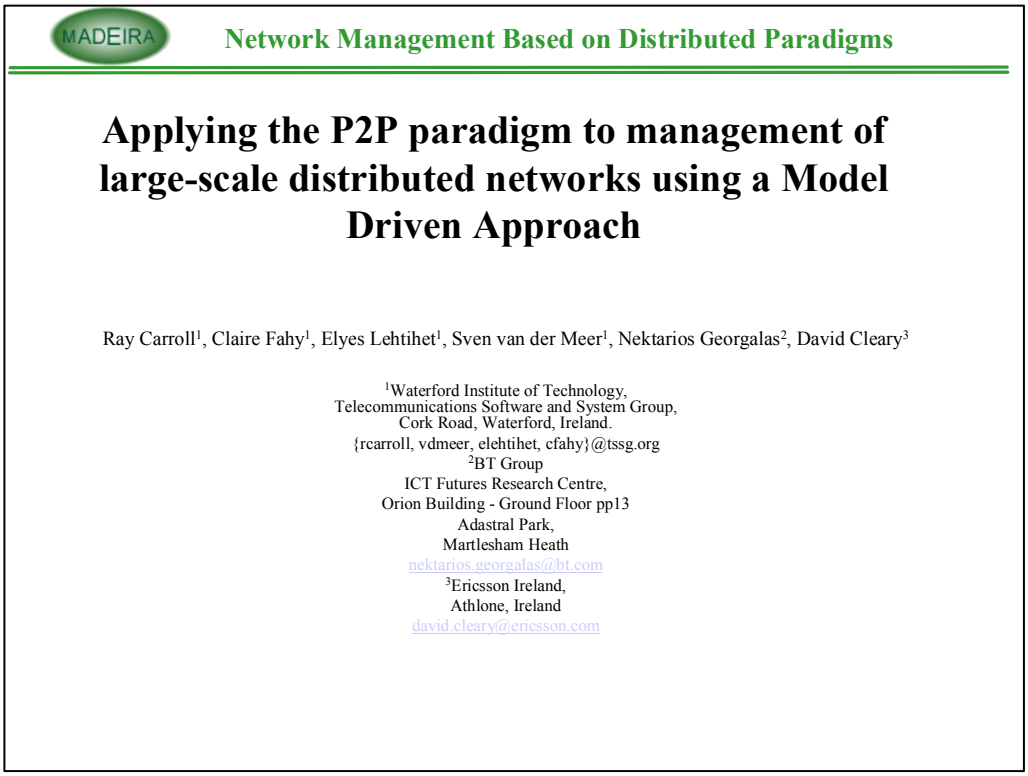


Applying the P2P paradigm to management of large-scale distributed networks using a Model Driven Approach

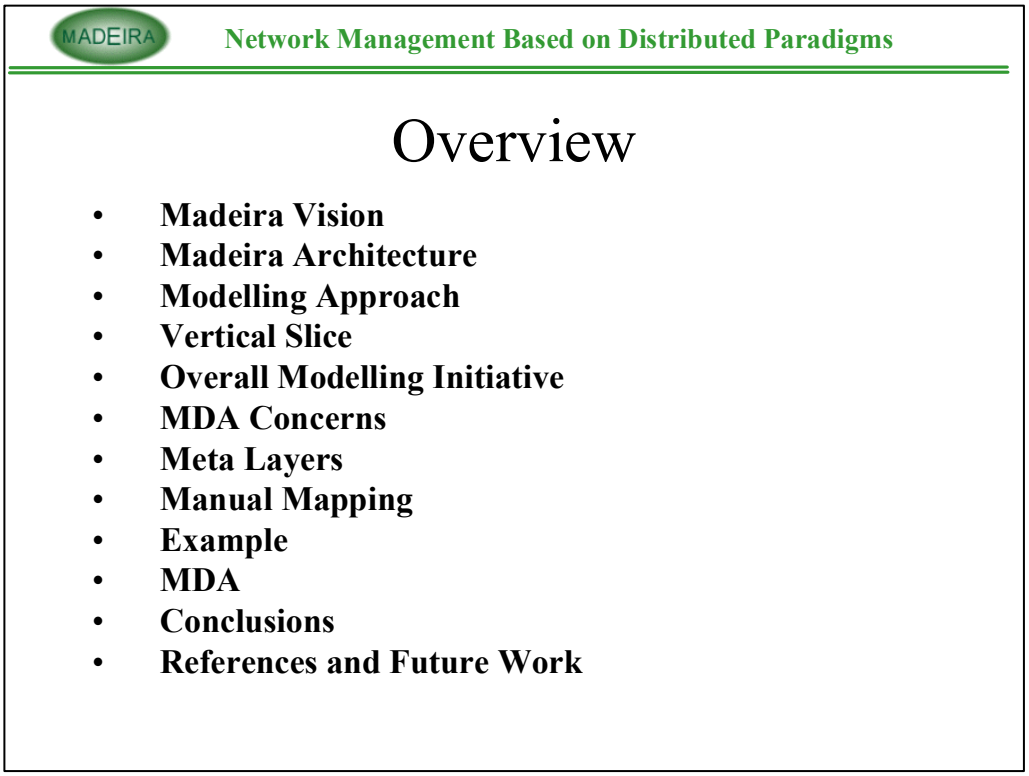
Ray Carroll[1], Claire Fahy[1], Elyes Lehtihet[1], Sven van der Meer[1], Nektarios Georgalas[2], David Cleary[3]

[1]Waterford Institute of Technology,
Telecommunications Software and System Group,
Cork Road, Waterford, Ireland.
{rcarroll, vdmeer, elehtihet, cfahy}@tssg.org
[2]BT Group
ICT Futures Research Centre,
Orion Building - Ground Floor pp13
Adastral Park,
Martlesham Heath
nektarios.georgalas@bt.com
[3]Ericsson Ireland,
Athlone, Ireland
david.cleary@ericsson.com

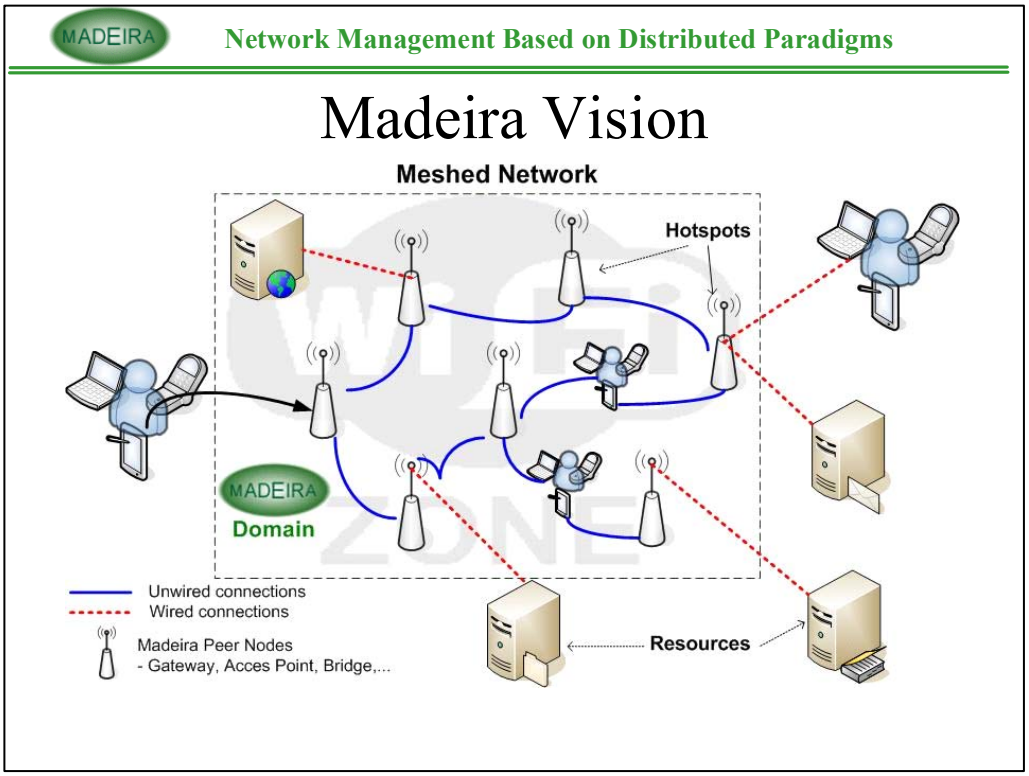
Abstract

This paper details an application of model-driven development undertaken within the Celtic initiative project Madeira. The objective of Madeira is to apply model-driven approaches to investigate large-scale distribution techniques in network management. So far, the project has concentrated on building a prototype system using the peer-to-peer (P2P) paradigm. For this one of the major challenges was to provide a model supporting P2P characteristics, such as 1) self-organisation, 2) symmetric communication and 3) distributed control and domain specific concepts for distribution and management as can be found in telecommunications. Our modelling approach had to consider the views of the different participants, such as equipment vendor (Ericsson and Siemens), network operator (BT) and service provider (BT, Telefonica). To minimise the complexity of the model, we have focused on fault and configuration management in a dynamically forming network of transient elements.

This paper will explain the complexity of the networks we consider and of the management tasks we have to cover. This can be briefly characterized by comprising a multitude of different, sometimes proprietary, technologies and diverse business models. We will motivate the application of new paradigms, in our case the P2P paradigm, to simplify management for seamless service provision to customers. Based on this, we will provide a case study of how we applied a model-driven approach to capture the complexity of the task and the complexity of the management activities, which ultimately led towards the specification of management information and behaviour of network nodes. The methodology we use is presented in the form of a 'vertical slice', where a logical portion of the project, of limited scope and functionality, is brought from the meta-level right through to the development stage, covering all modelling and architectural work as well as the underlying platform aspects.



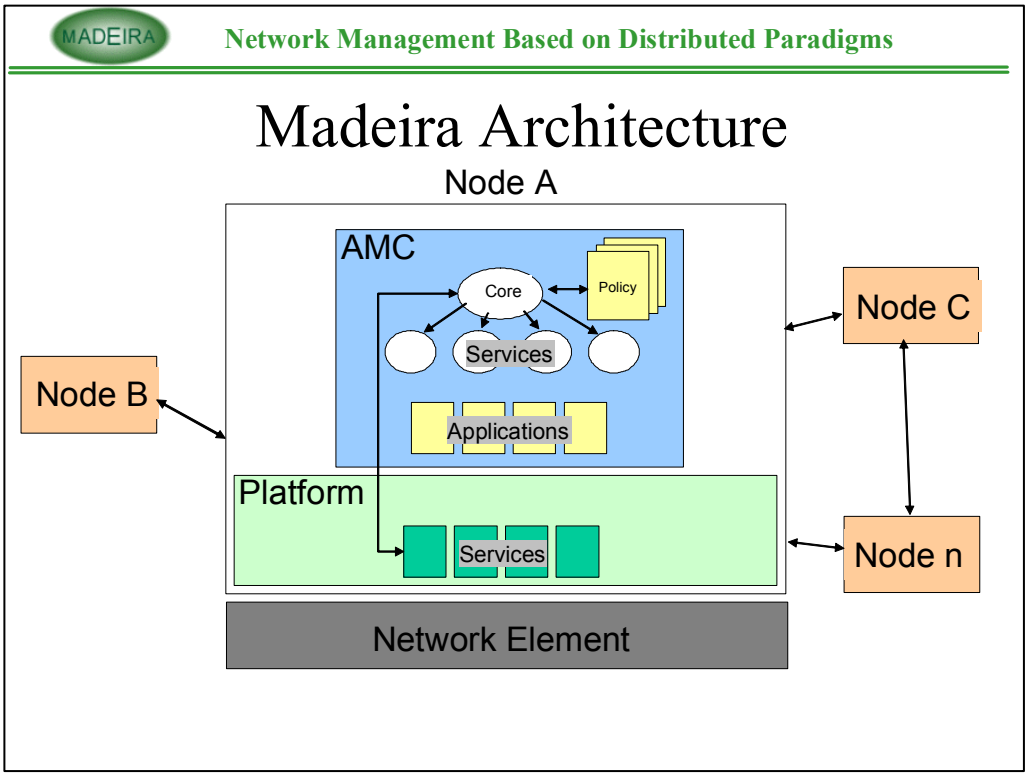
This paper's aim is twofold. Firstly we will describe our experience in adopting a model driven approach for the system design using manual mappings between modelling layers. Secondly we will then present our findings from attempting to, in parallel with the manual approach, adopt an automated model driven approach through the use of MDA (Model Driven Architecture) compliant tools. Additionally we will also describe some of the difficulties experienced in adopting the model driven approach in a multi-partner European project.



Telecommunications networks will evolve to contain very large numbers of nodes involving networks of networks (with more operators, equipment vendors and owners) and with larger technological diversity and heterogeneity of elements. The requirements placed on management systems will lead to new paradigms being needed in order to deliver more adaptive and simpler systems, particularly required will be a new relationship between Configuration Management and Fault Management functions. Key requirements on the management systems are that they are capable of self-configuration within a network, capable of inter-network collaboration for management tasks, are much cheaper compared to today (i.e. lower Operating Expenditure (OPEX)), more adaptive to service requirements, varying network deployment scenarios and highly usable in all respects.

The objective of the Madeira project [1] is to investigate large-scale distribution techniques in Network Management and ultimately to provide novel technologies, namely P2P concepts, for a logically meshed Network Management System (NMS) that facilitates dynamic behaviour of transient network elements. This will enable self-managed services and network elements of increased scale, heterogeneity and transience thereby reducing OPEX. In particular, this research aims to develop a logically meshed distributed non-hierarchical telecommunications management system that focuses, within a case study, on the inter-related configuration and fault management (CM/FM) aspects of a network and its element in large-scale dynamic heterogeneous networks. It will also consider the relationship between a network management system and an operators operations support system (OSS) in such an interlinked CM/FM scenario. It will take a scenario based user-centric approach to requirements definition and emphasise the services that the system elements shall offer from the perspective of the users (i.e. Operators and Application Developers).

The focus of this project can be divided into 3 core areas. These are: computing and communications platform; data modelling and architectural requirements. While these three areas are intrinsically linked this paper focuses specifically on the modelling concerns and on how the application of the model-driven approach in this context can benefit the development of such a management system.



Based on the approach of applying P2P concepts to the management domain the Madeira architecture has been designed with a number of key principles in mind. The most important of these principles is 'heterogeneity' or the ability of the Madeira system to be applied to many management domains and across heterogeneous devices and platforms. The main vehicles for this generic management is the usage of policies, notifications and applications.

The Madeira architecture is essentially composed of an Adaptive Management Component (AMC) and a Platform. The AMC is the component that manages a given node and together many AMCs can orchestrate the overall behaviour of the meshed network. These AMCs have the ability to exchange and export Network Management information between peer management applications and are deployed as an overlay network, communicating using the peer-to-peer paradigm. The AMC itself is composed of a number of sub-elements which facilitate this management.

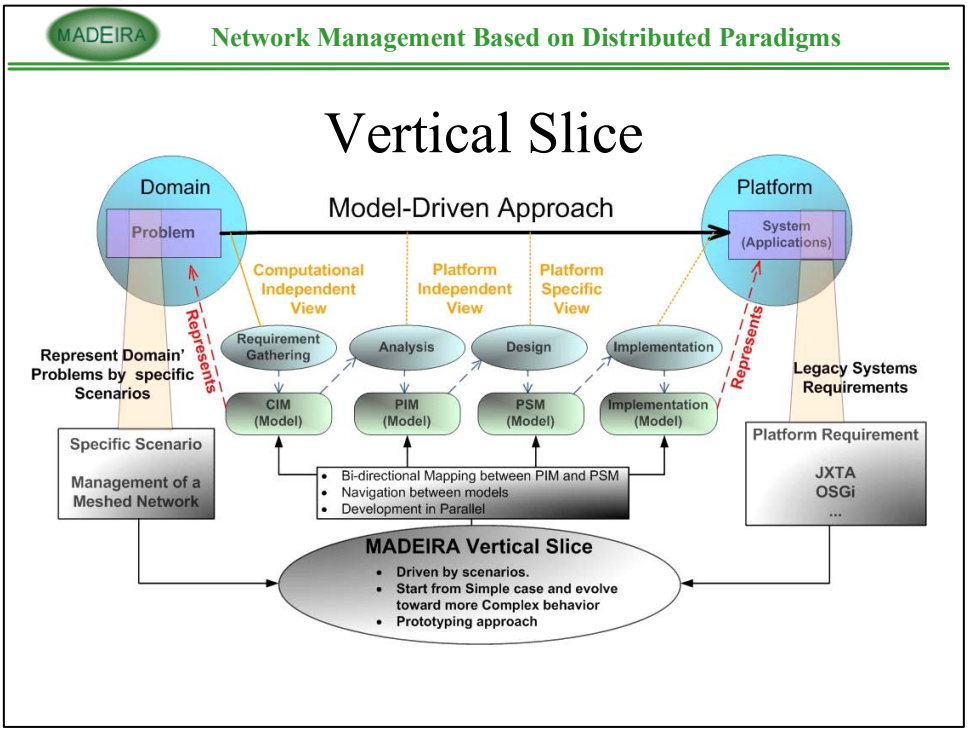
The **AMC Core** is the is the primary component or 'brain' of the AMC and, based on notifications and policies, orchestrates the services and applications to facilitate the required network management function. **Services** are components that provide some functionality required by the AMC Core. **Applications** provide the actual management functionality specific to a particular device. Applications are physically divided into parts that run within each AMC but are logically connected through peer interactions. For example each AMC will run some Fault Management application which together with FM applications running on other nodes constitutes the overall FM application of the entire meshed network. **Policies** provide the generic management functionality shared by all AMCs within the same management domain. The generic management specified in policies is then mapped to applications. **Notifications** enable inter-AMC communication, via the Madeira platform, about events within the mesh network and also provide a means to a logically distributed applications.

Modelling Approach

- Vertical Slice
  - Cross-section of the modelling aspects of the project for a specific scenario.
  - Useful to evaluate our approach.
- Adopt Model Driven Approach
  - Use MDA to model vertical slice.
  - Many concerns among consortium.

There are a number of key decisions that were made in terms of the approach taken to modeling in this project. The first decision was to adopt a 'Vertical Slice' approach. The vertical slice is essentially a full cross-section of the modeling and architecture aspects of the project for a specific scenario. The purpose of this is to evaluate our approach, highlighting problems or issues, which can then be addressed and re-worked. It also serves to establish requirements towards the Platform.

Clearly the second decision was to investigate the application of MDA within this vertical slice. This in itself has lead to many interesting issues arising as a result of a consortium of pan-European partners with divergent concerns. These concerns and the overall conclusions drawn in regard to these are outlined later in the presentation.

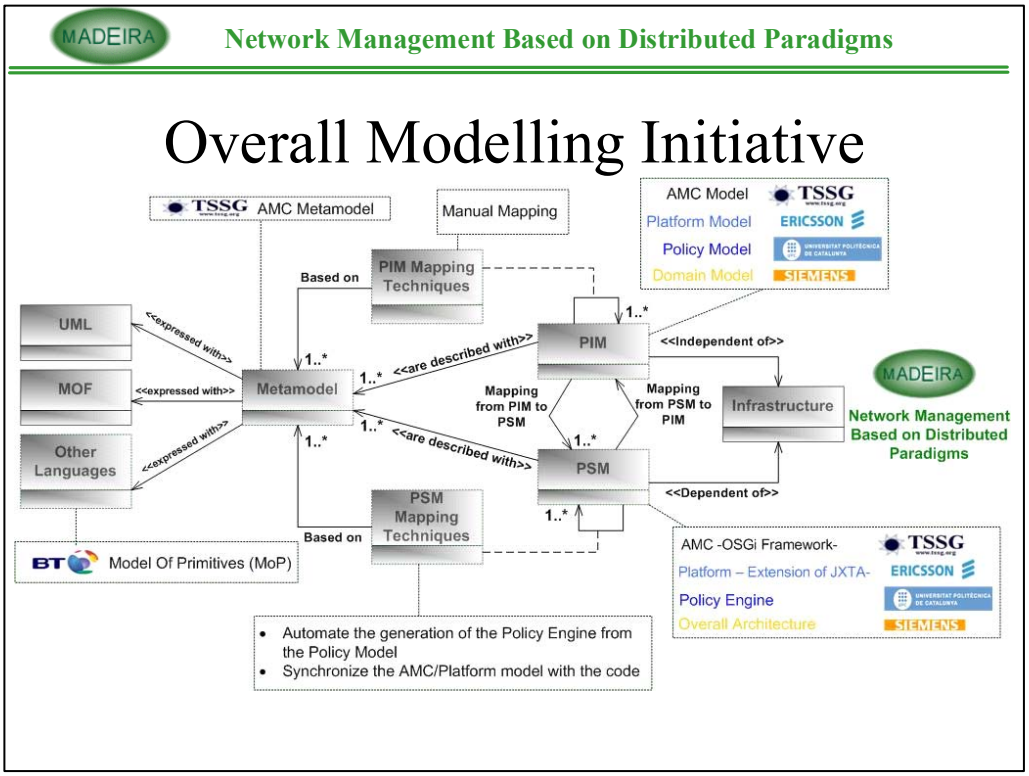


The system development lifecycle process is a problem-solving process where requirements maybe considered problems occurring within a domain, systems that address the requirements may be considered solutions residing within an environment, and problem solving involves understanding a problem, solving the problem, and implementing the solution [2].

The diagram above shows how both MDA and the Vertical Slice concepts fit together. The aim of the vertical slice approach is to develop the platform independent models around a specific scenario. Instead of trying to tackle all aspects of the problem domain, we instead select a set of requirements and design our solution based on this chosen subset. This subset then provides the initial input to our Platform Independent Model (PIM). The chosen scenario was orientated around a basic configuration and fault management use case where nodes must first configure themselves, then identify a fault and re-configure based on the identified fault.

Based on the architecture described earlier we saw that applications are considered as separate logical functions from the underlying AMC, where the AMC provides the supporting infrastructure for these management applications. As such the vertical slice scenario drives the specific requirements of both the AMC and the Applications themselves. Also it should be clear that, as the supporting infrastructure, the AMC is a pre-requisite of the Application and as such the vertical slice must first address the AMC model. The relationship between the application models and the AMC model is not a linear one, as to design the AMC we must consider its applications yet the applications are inherently reliant on the AMC. This is the main aim and strength of the vertical slice approach as it will rapidly produce a prototype and identify major issues and areas of concern.

With this PIM we can then generate a Platform Specific Model (PSM) based on mappings to our chosen platform. To represent the platform specific model, we started by identifying the necessary components of our system, as defined by the original scenario, and the platform requirements for implementing a P2P network management system. Once a stable representation of the basic functionality has been developed we can then build extensions that consider all other aspects of the entire problem domain. This is an iterative process where every model is documented, reviewed and updated by the partners according to their area of expertise.



The case of a vastly distributed, non hierarchical, logically meshed peer-to-peer management system for heterogeneous networks generates a set of models that try to capture the complexity of the P2P Network Management System. The slide above highlights the overall modelling initiative and the partition of the work between the partners. Every partner, in their area of competency, developed specific models in coordination with the other partners.

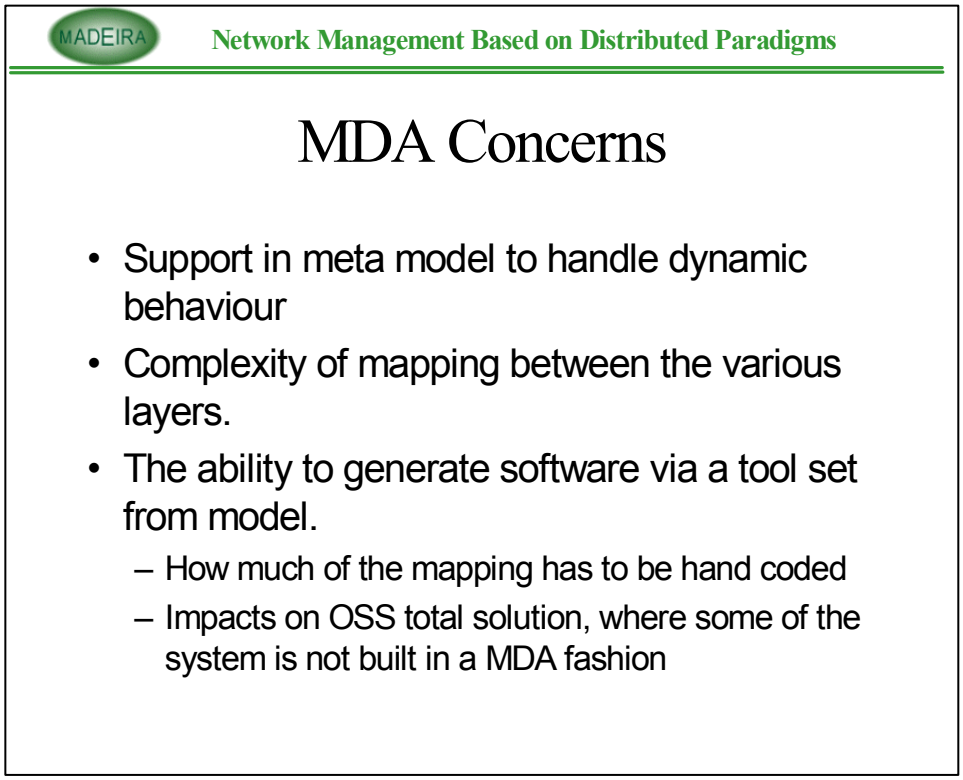
Some important points are:

•Mappings from PIM to PIM have been done manually

•We are investigating tools and process to automate the mapping from PIM to PSM and PSM to PIM

•Having the underlying infrastructure specification is necessary to specify the mapping.

•The Mapping itself is inspired by the best practises in software engineering. In our case it covers different areas: P2P Networks (Protocols and APIs) and Policy-based management.



MDA Concerns: Using MDA to Build a Distributed Management System.

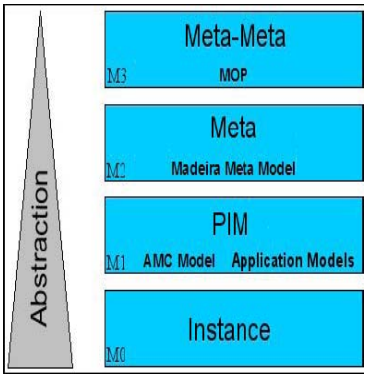
As described previously, the Madeira project is concerned with management of dynamic radio networks. Our approach is one centred on the adoption of meta-modelling principles. Our problem offers many difficult challenges and differs from the current adoption of model driven development. The main issues we face are concerned with the management of a loosely coupled dynamically forming communication platform [3].

If we analyze the conceptual approach of MDA, which is at the core of meta-model [4] thinking, it clearly illustrates both the strengths and weaknesses of the MDA approach. From our experience it allows you to think of the problem you want to solve in a different way. You can abstract the problem and break it up into smaller parts, which can be solved individually. It gives you a way to visualise the problems and to solve them without being swamped in the immensity of all the issues that need to be solved. This is extremely important and is why all modelling is useful and used. However there is another saying that illustrates the problem with MDA just as succinctly, *"The devil is in the detail"*. This is true of almost all software systems, however in the field of distributed real-time network management it is even more so. The consequence of this means that while it is very important that the high level modelling and design is done correctly it is equally vital that any design/design methodology is capable of dealing with minute detail.

Two noticeable problems with the MDA theory arise from this assertion. Firstly in order to deal with the detail the model becomes more and more complex. This may be unavoidable but it can lead to the loss of the advantage that the model gave you in the first place, i.e. a level of abstraction. Secondly changes made to the PSM in order to account for the details that are abstracted out of the PIM are not reflected back into the PIM, which over time makes the PIM increasingly irrelevant and not better than standard textual descriptions (possibly even worse as it becomes more expensive in tools/expertise to maintain).

Meta-Layers

- M3-Meta-meta Model
  - Model of Object Primitives
- M2 – Meta Model
  - Management and Middleware Architecture
- M1 – PIM (AMC Model)
- Instance



As specified by the MDA we adopted the MOF 4-layered architecture as our overall modelling framework. At the same time, and in order to address the MDA concerns of the consortium, we added more detail in the meta-levels. By doing this we aimed to ease the mapping between layers and also to ensure that we addressed the problem of capturing behaviour within the meta-layers.

At the top level is the Madeira meta-meta-model (M3) which was based on the Model of Object Primitives [5]. The MOP provides a basic set of model concepts on which the other model layers could be built. Key to MOP is that it incorporates policy and events so behaviour is considered from the outset, at the highest level of abstraction.

At the M2 equivalent level is the Madeira meta-model. Here the modelling constructs defined in the meta-meta model begin to become influenced by the problem domain, incorporating concepts that are useful to describe our particular system. The Madeira meta-model is based on work carried out on the Middleware and Management Architecture (MAMA) [6]. This premise of this work was to integrate the 'best of breed' approaches to management and middleware in an attempt to define a unified management and middleware architecture. As such MAMA adopted concepts from prominent distributed processing and management standards/approaches like the RM-ODP, TINA-C, CORBA and CIM. While this was adopted as the basis for the Madeira meta-model, it needed to be extended to incorporate concepts for a peer to peer management paradigm, in line with the concerns expressed previously. As a result Connections, Transactions and Distributed Application elements were added to the meta-model to represent P2P network management concepts that were key to the Madeira scenario.

The AMC model utilises the newly defined concepts within the preceding meta-layers to describe a single Adaptive Management Component. One of the key aspects of this is the representation of policy, notifications (events) and applications in such a way as to describe the management mechanism of a Madeira node.

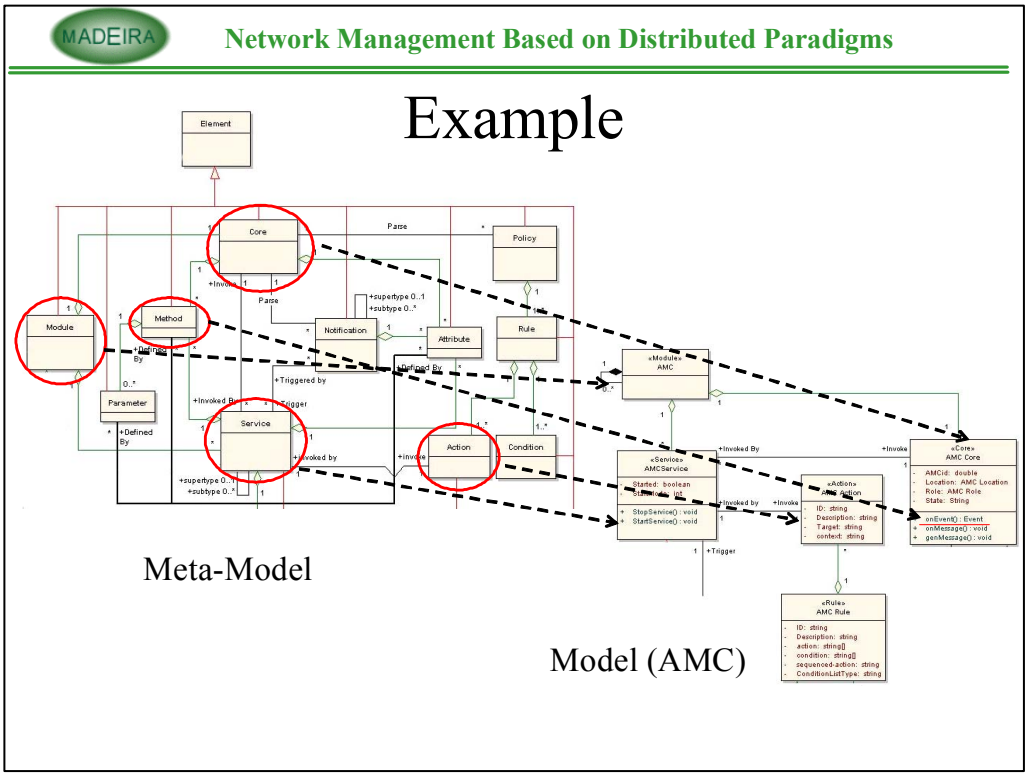
Also at this level there are a number of Application Models. As the AMC is a generic tool for applying any management function to the network we needed to define specific models of the management application. These models were based around the vertical slice scenarios, specifically Configuration and Fault Management.

Manual Mappings

- Entirely UML based
- 'Simple' UML modelling tools
- No standard mappings.
- No formal mapping between layers
  - Simple stereotypes
  - No control mechanism for meta-model conformance
- Hand-coded

As a result of the MDA concerns expressed within the Madeira project two separate approaches to the modelling were adopted. The approach descried here is best described as a Manual Mapping Approach. The manual approach describes the method of mapping between the meta-layers using only a 'mental-model' and 'simple' modelling tools. This essentially involves creating each model based on it's preceding meta-model using simple UML stereotypes. As a result there is no formal mapping between layers and hence no strict means of ensuring that the each model conforms to its meta-model. Also with no formal mappings defined then it is very difficult to maintain consistent models as each developers mapping can vary. Using this approach the models defined essentially become the primary design specification and are used to hand-code the Madeira system.

We began our Vertical Slice approach using the basic AMC model as this is a pre-requisite for all the application models. Each model element in the AMC was stereotyped from meta-model elements as described in the previous paragraph. An example excerpt is shown on the next slide.



In this example we can see that elements from the Meta-model on the left are mapped to the model on the right using basic UML stereotypes. So for instance a Module in the meta-model contains one core object and many service objects, where both Service and Core contain Attributes and Methods. A Service is invoked by a policy Action where the Core parses a Policy. In the AMC model we can see that an AMC is a Module and as such contains an AMC Core and AMC Services. AMC Core and AMC Service have attributes and methods.

The model here is too flexible and cannot be automatically be verified and constrained. For instance the AMC as a Module strictly speaking should have only one Core and any number of Service however using stereotypes there is no way to verify that this is actually the case. This approach is error-prone and is only useful as a roadmap for developers to build the software, often requiring much manual validation and correction. Simply put this approach is not formal enough.

MDA

- Formal mappings
  - validation of conformance is inherent
- Inheritance of constraints and behaviour
- Standard representation of mapping
  - More consistent models
- Possibility of defining grammars and generating code and parsers.

The alternative approach is to use an MDA tool to develop the models where each meta-level becomes an instantiation of the previous meta-model. In this way stereotypes become not only text-based typing but physical instances of a related meta-element. This allows constraints (and behaviour) expressed in the meta-model to be inherited by subsequent instantiations of this model. Formal standard mappings ensure that the models built are consistent across the development lifecycle. This approach can provide huge advantages in software development as it allows early testing and validation of models before implementation as well as more consistency and conformance. One of the main benefits of MDA is obviously the generation of code. As the model is formally represented in a way that can be machine manipulated it is also possible that language grammars can be easily built from that model and that resultantly the development of parsers can be simplified.

In this case we started by specifying the MOP model in our chosen MDA tool. This provides the basic building block upon which all the modelling work would be built. When specifying the Meta-Model the MOP-based meta-meta model is then used as it's meta-package. This enables the meta-model to be a direct instantiation of the MOP meta-model and hence inherit its properties and constraints. Verification of this model is automatically carried out by the tool as any invalid instantiation of a meta-meta model element will not be allowed. However while the manual mapping approach allowed us to reach a much greater level of maturity in our models; we were not able to achieve a similar result with the MDA approach. This was due to a number of reasons which are detailed in our conclusion.

Conclusions

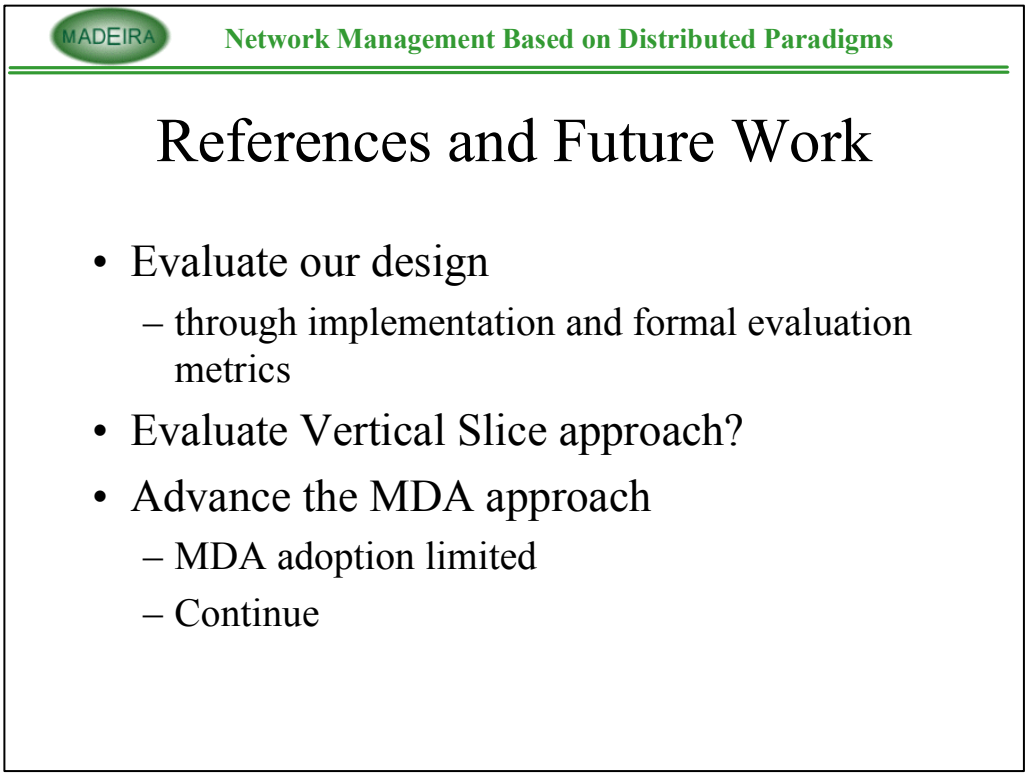
- MDA holds a lot of potential
  - Model instantiation and verification
  - Constraint inheritance through layers
  - Grammars and Parsers
- MDA tools too immature
  - Unstable
  - Not able to model some simple constructs properly
- MDA Standards are still being developed
- Domain specific elements in meta-layers
  - More expressive modelling constructs
  - Instinctive mappings

MDA tools, like MDA itself, demonstrated extraordinary potential. However at this early stage in the evolution of both there still remain a number of issues.

So far our experiences of many of the existing MDA tools is that while much progress has been made they are still very immature in their nature. Also the ability to represent some basic modelling constructs was lacking in some of these tools which presents a major obstacle to system developers/modellers.

Also a number of the key MDA standards are still under development. MOF and UML are standards of the OMG and all the Modelling tools implement these specifications. For model transformation (validation, checking consistency, etc.) the important OMG standard is QVT (Query View Transformation). The final specification of QVT is going to be finalized, and publicly available, by July 2006. Similarly for MOF version 2.1. The compliant tools for these core technologies should be available next year. So the available tools are not mature because of the non-finalized standards and lack of cases studies.

Adding domain specific elements within the meta-layers resulted in a much more expressive and instinctive meta-language for the development of the Madeira system. This also made the mapping process between layers much less painstaking as the required mappings became much more obvious. In particular introducing behaviour primitives at an early stage ensured that at all stages of development system behaviour was considered.



The next stage of this work is to validate the overall Madeira vision by implementing the specified system and devising formal metrics on which this system can be evaluated and compared to traditional distributed management techniques. Another task which we are planning is to carry out a qualitative review of the vertical slice approach. This review will attempt to ascertain whether the vertical slice approach provided significant benefits to the development of the system.

The work we have been able to in terms of MDA has been limited. Our ultimate aim would be to use MDA to generate code based on the models provided and then to evaluate this code against the hand-written code. This we feel would provide an extremely useful metric as to the current state of MDA. Most of the code generation is linked to the sequence diagrams for Fault Management and Configuration Management. We used stereotypes and constraints but we mapped to real world objects manually since there is no final implementation of the OMG standards. We still investigating the potential of many commercial tools (Rational XDE, Borland Together, SparxSystem) and free plugins for Eclipse (IBM Model Transformation Framework, IBM Design Pattern Toolkit, MerlinGenerator).

References

[1] The Madeira Project, http://www.celtic-madeira.org/

[2] Sinan Si Alkhir, “Understanding the model driven architecture”, published in “Methods & Tools” October, 2003, an international software engineering digital newsletter published by Martinig & Associates. http://home.comcast.net/~salhir/UnderstandingTheMDA.PDF

[3] Martin Zach, Daryl Parker, Liam Fallon, Christian Unfried, Miguel Ponce de Leon, Sven van der Meer, Nektarios Georgalas, Johan Nielsen. “CELTIC Initiative Project Madeira: A P2P Approach to Network Management”, Eurescom Summit 2005 Ubiquitous Services and Applications Exploiting the Potential. 27 - 29 April 2005 Heidelberg, Germany

[4] Model Driven Architecture (MDA), MDA Guide Version 1.0. Copyright © 2003 OMG, http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf

[5] Nektarios Georgalas, “The Model of Object Primitives (MOP)”, Succeeding With Object Databases: A Practical Look At Today's Implementations With Java™ And Xml, October 2000, 464 pages, ISBN 0-471-38384-8

[6] Sven van der Meer, “Middleware and Application Management Architecture”, PhD Thesis, Berlin, Germany, September 25, 2002. http://edocs.tu-berlin.de/diss/2002/vandermeer_sven.htm