



**Bournemouth  
University**

# Embedding Requirements Within the MDA

Ali Fouad – Postgraduate Researcher.

Software Systems Research Centre

Bournemouth University, Poole, Dorset, BH12 5BB.

<http://www.bournemouth.ac.uk/ssrc/>

# Introduction

- Research Overview
- MDA
- xMDA
- RAD
- Demonstration
- Questions
- Contributions
- Conclusions

*“To what extent can the MDA incorporate a requirements definition created by business user involvement within the CIM phase of the MDA to be practical in the development of software systems?”*

# Research Overview (1)

- Aim 1: To examine the definition of the CIM within the MDA and consider the appropriation of it as an interface with the business user for defining requirements in MDA notations.
  - Objective 1: Examine the connection between the MDA and business.
  - Objective 2: Determine the sufficiency of the CIM at delivering requirements to the MDA.
- Aim 2: To discover how other modelling techniques which are accessible to the business user, might be integrated with the MDA in terms of method and notation, with the focus on transformation and traceability.
  - Objective 3: Investigate how requirements can be supported by the PIM.
  - Objective 4: Examine how useful the CIM is at describing requirements.

# Research Overview (2)

- *Aim 3: To extend the framework of the MDA to account for specification within the CIM.*
  - *Objective 5: Derive mechanisms to adequately capture and realise requirements within the MDA.*
- *Aim 4: To determine the academic and commercial value of extended mechanisms.*
  - *Objective 6: Verify the proposed mechanisms to extend the MDA.*

- OMG (2003) *MDA Guide* Version 1.0.1
- Computation Independent Model (CIM)
  - Problem domain and Requirements
- Platform Independent Model (PIM)
  - Class Diagram
  - Activity Diagram
- Platform Specific Model (PSM)
  - Code
- Notation : UML (support for BPMN)



**Bournemouth  
University**

# MDA Literature - Themes

- **Dominated by notations derived from Software Engineering paradigms**

The MDA vision has yet to be applied to concepts in RE (Kabanda and Adigun 2006)

- **Semantic gap (Elliott and Raynor-Smith 2000)**

The UML “falls into the cracks between technical people (developers, architects) and non-technical people (business analysts, project managers, etc). [The] UML is too technical for non-technical people, and not technical enough for technical people” (Ford 2009)

- **Limited information available on the practical use of Requirements / BPM methods, specifically in the context of the MDA**

The MDA does not include any “precise rules or guidelines explaining how Software Engineers can use” the CIM, PIM and PSM (Garrido et al. 2007)

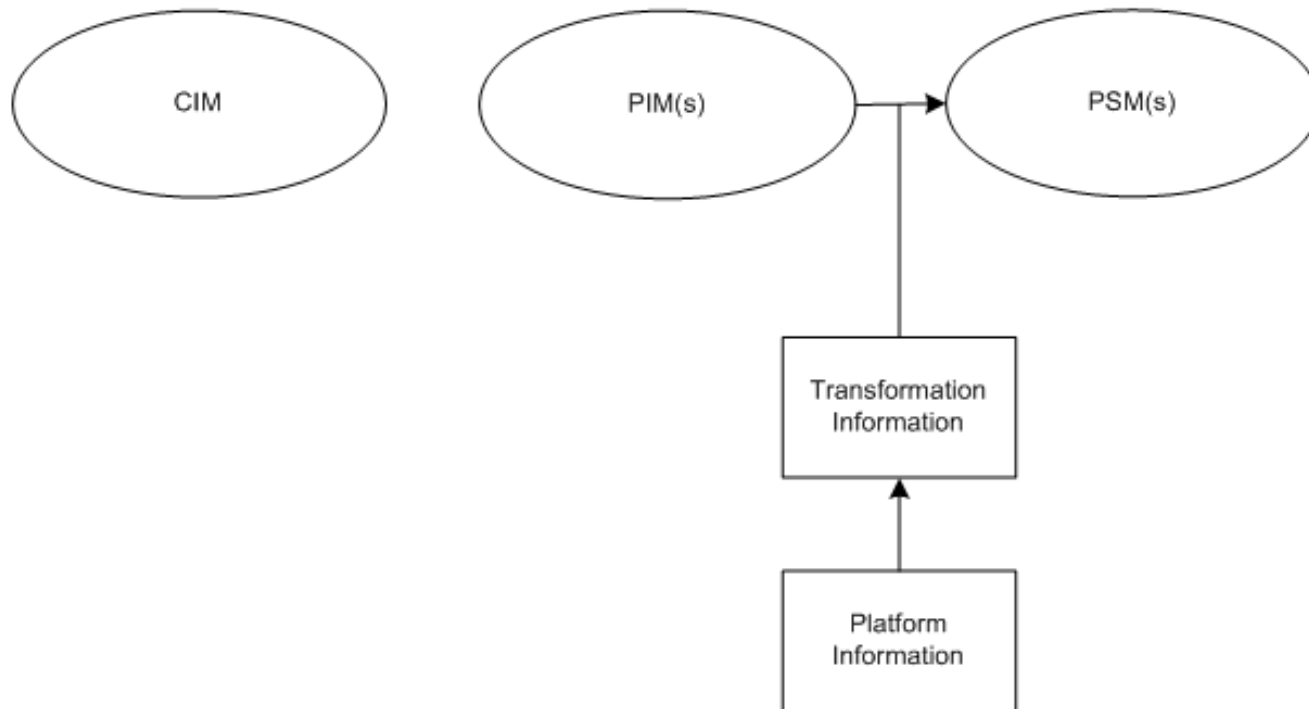
- **Neglect of specification in CIM phase in terms of notation and method**

The CIM “merely informs the decision makers about the system’s context but does not influence design decisions in a functional describable way” (Karow and Gehlert 2006)

- **Mature technology?**

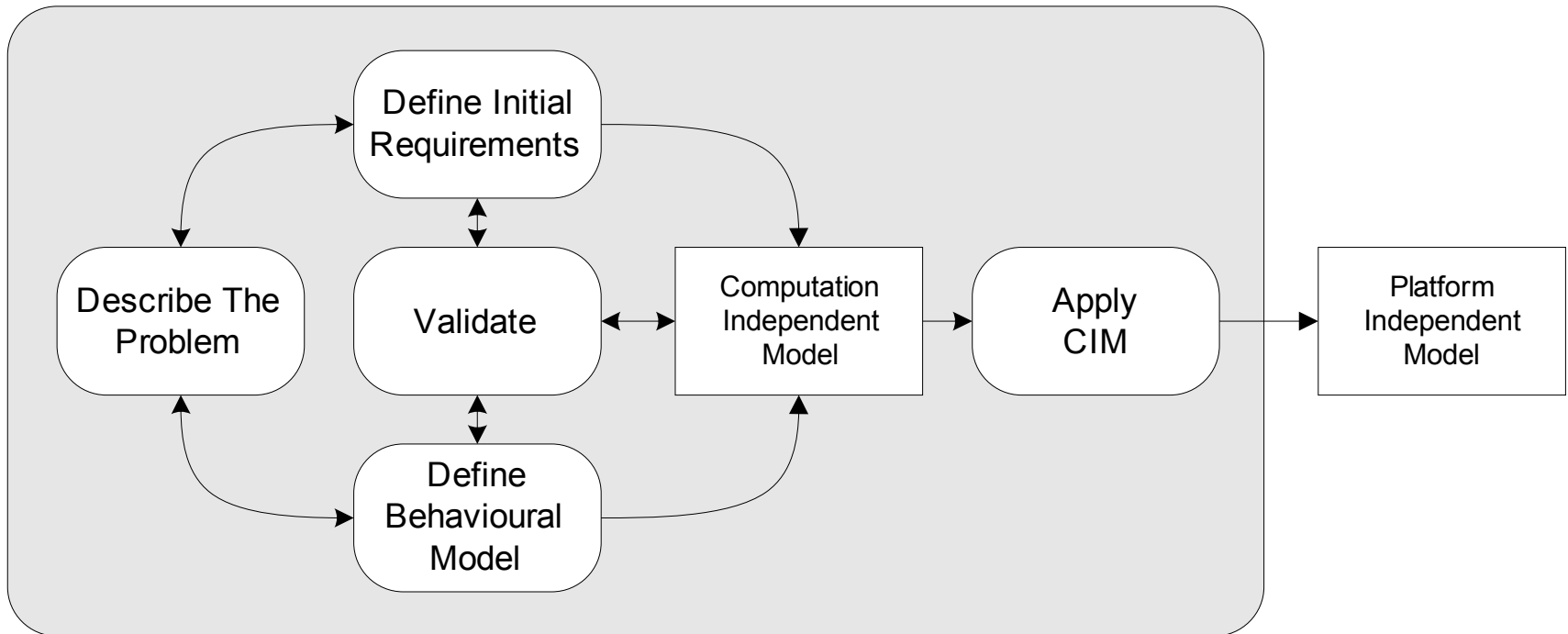
Transformation engines such as Eclipse QVT Declarative, Medini QVT, ModelMorf and MOMENT-QVT are still in the development stages (Bureck 2009; Kusel et al. 2009)

# MDA – Current State



*MDA Viewpoints (Source: developed from (OMG 2003))*

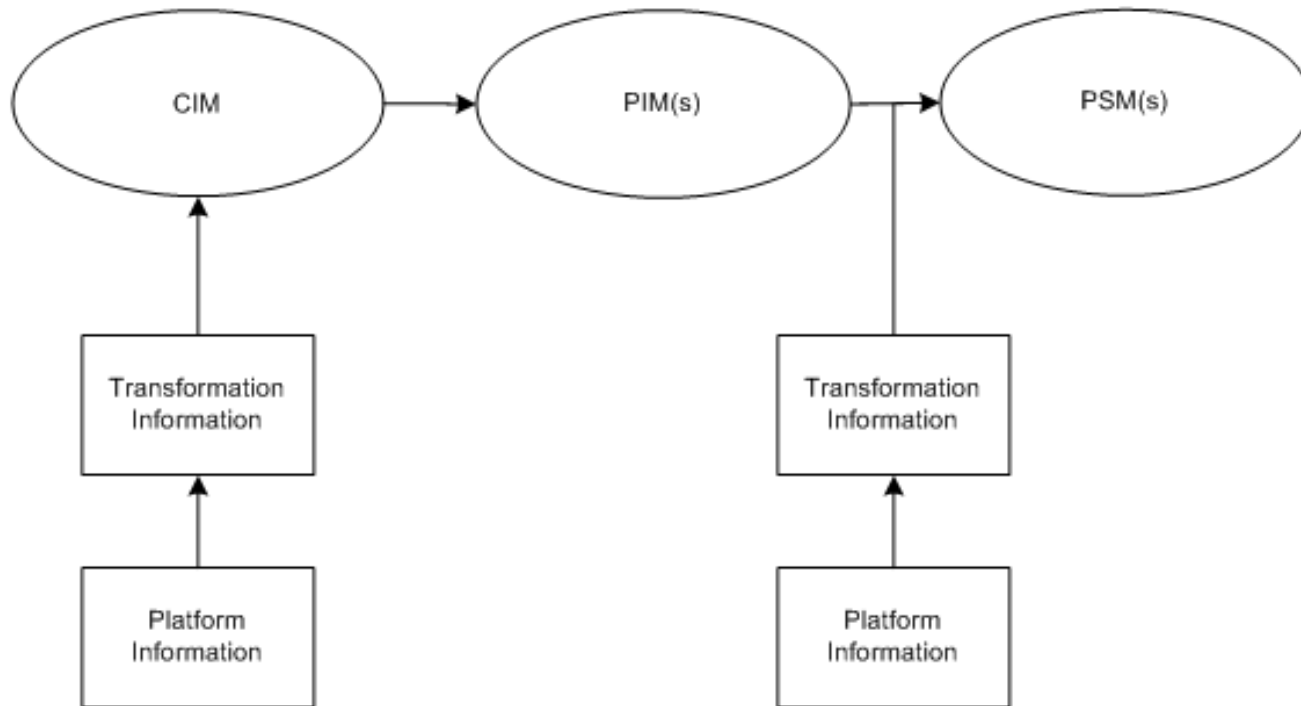
# MDA – Extension



*Requirements and specification included as part of the CIM definition*

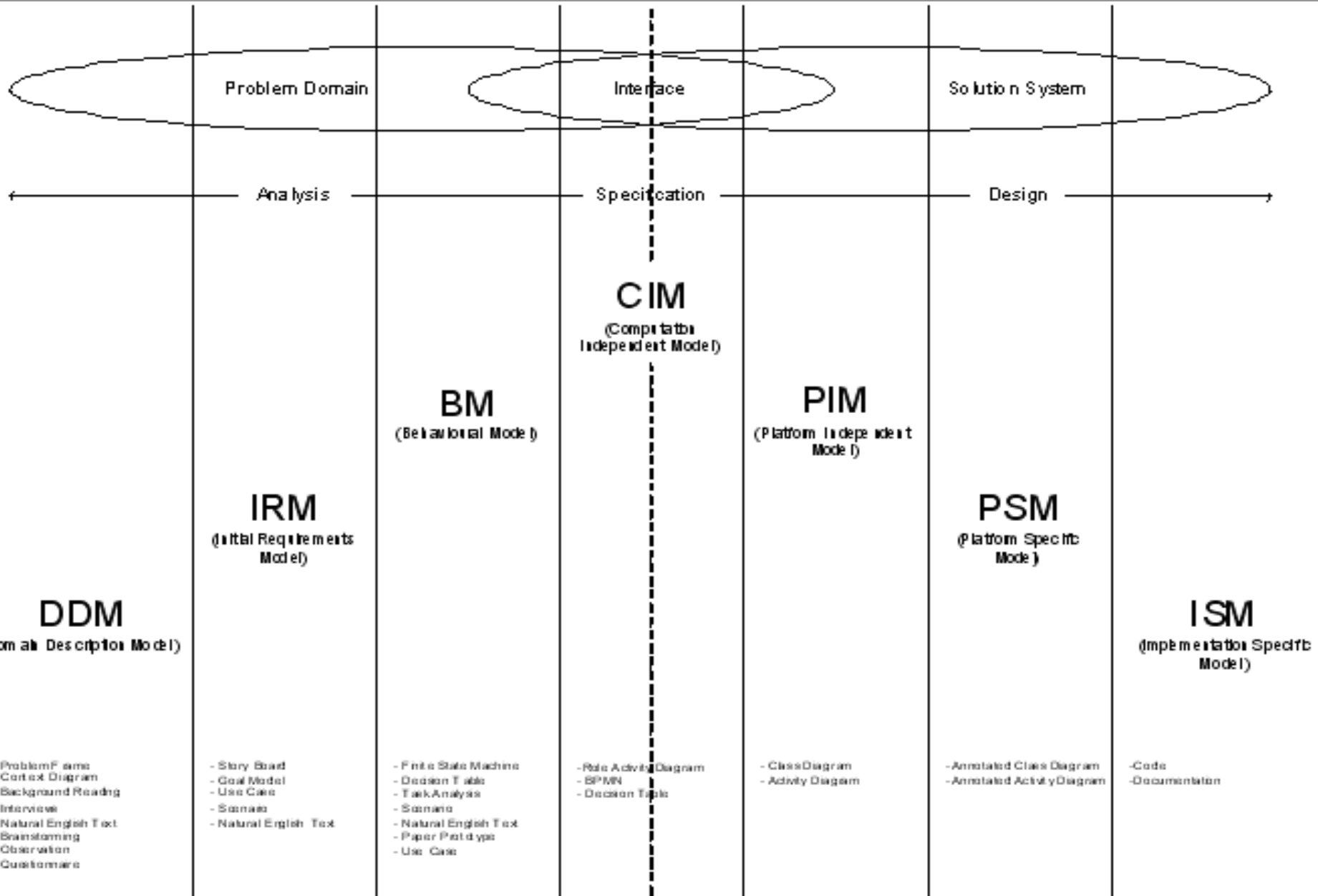


# xMDA – Framework Proposal

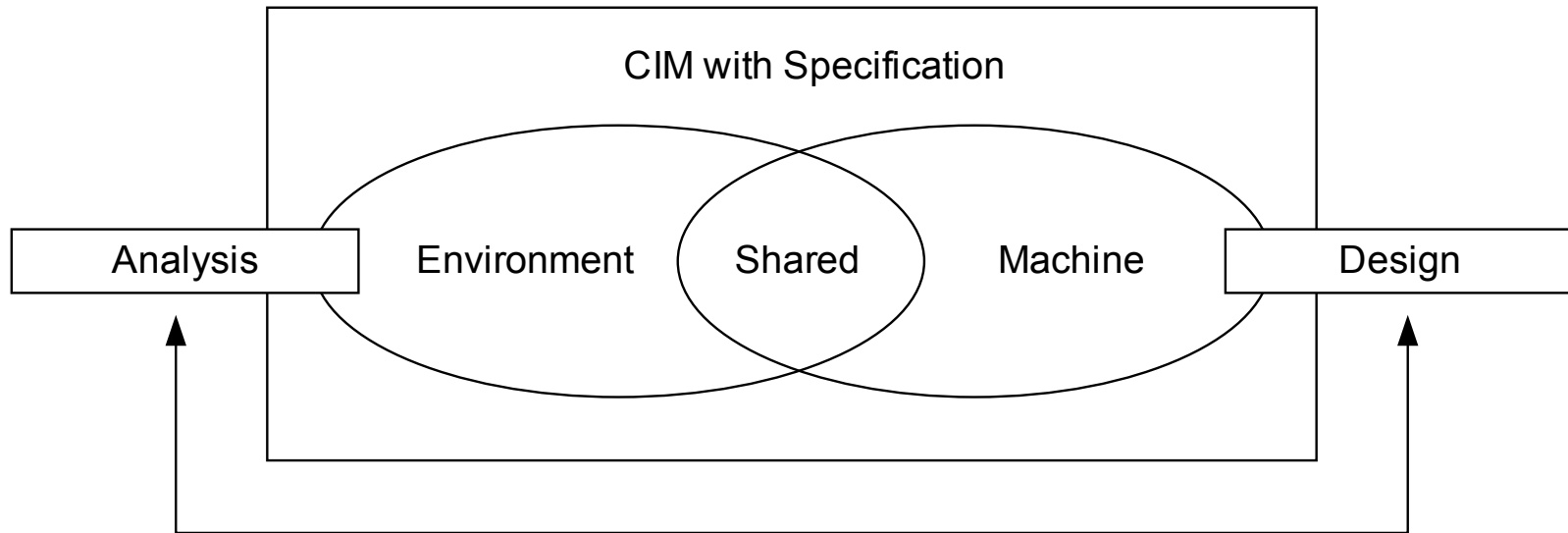


*Extended MDA (Source: developed from (OMG 2003))*

# xMDA with pre-CIM



# xMDA – Specification

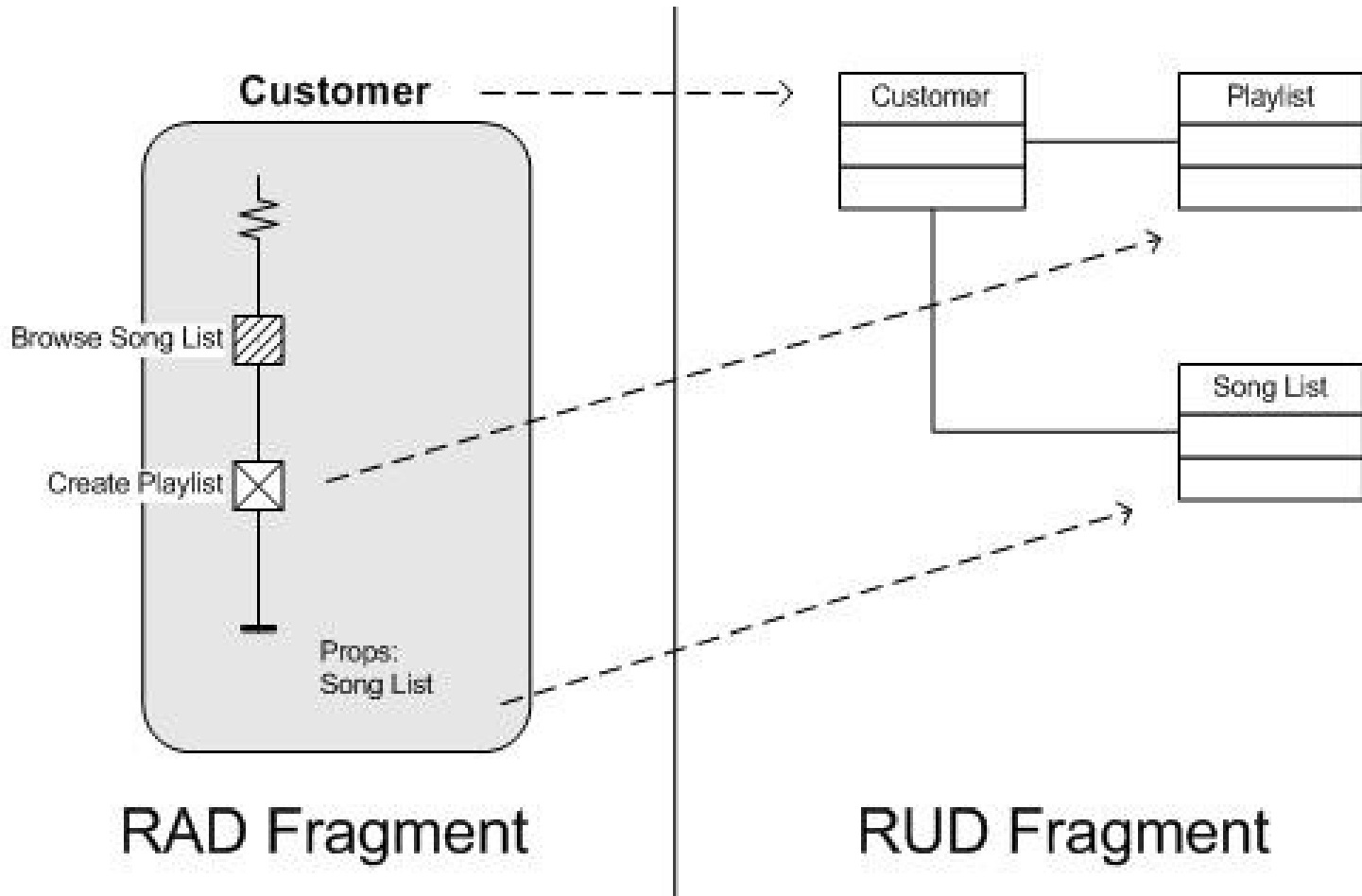


*The integration of the xMDA with Jackson's systems of prime concern.*

# Role Activity Diagram (RAD)

- Martyn A. Ould - Business Process Management: A Rigorous Approach (2004)
- RAD is firmly grounded in Business Process Management (BPM) – Roles and Interactions
- RAD with the MDA

# RAD - Situating Requirements within the MDA



*RAD fragment for jukebox example and associated Role Utility Diagram Fragment.*



Bournemouth  
University

# xMDA - Method Proposal

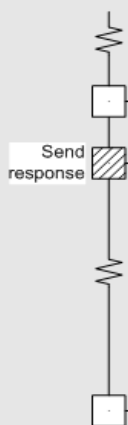
1. Define *Environment RAD* from analysis.
2. Derive *Shared RAD* to examine shared phenomena.
  1. De-couple environment concerns to create *Machine RAD*.
  2. Complete *Class Discovery* via *Tri-Step* analysis (optional).
3. Apply Transformation and Platform Information to create UML representations.
  - Use Case Diagram.
  - Activity Diagram.
  - Class Diagram.

# Demonstration – Follow Up Call Process

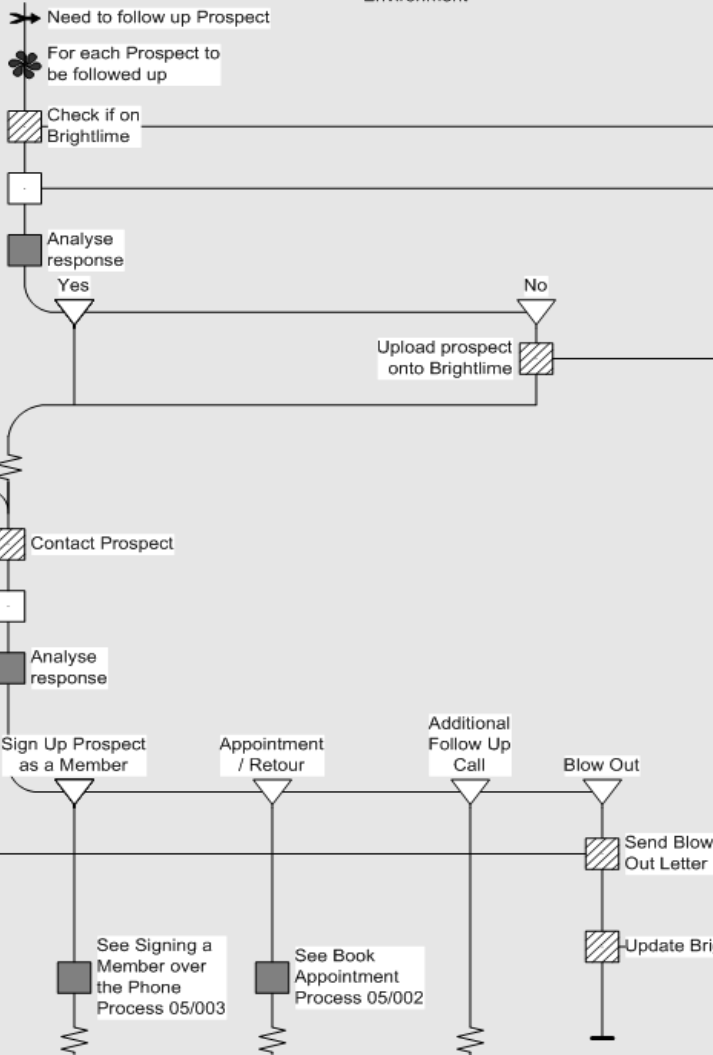
- A Sales Advisor is required to follow up on a potential member (or prospect) with the view to attaining membership to the club.
- Triggered by the need to follow up a prospect, the model illustrates that for each prospect to be followed up, the sales advisor must first check to see if the prospect's details have been retained on Brightlime (an existing prospect database).
- If the prospect details are not present, a new record is created.
- Once the details exist, the sales advisor is in the position to contact the potential client. Upon making contact, one of four routes can be taken.
- The prospect could be signed up as a member (this is catered for in an alternate process model - 05/003)
- An appointment or a retour can be arranged for the prospect to visit the club (process model 05/002)
- An additional follow up call could be required, in which case the same process is followed (illustrated by a looping line state)
- The sales advisor may blow the prospect out, by sending a letter and updating the prospect database.

# Environment RAD

## Customer ✓ Environment



## Sales Advisor ✓<sub>2</sub> Environment



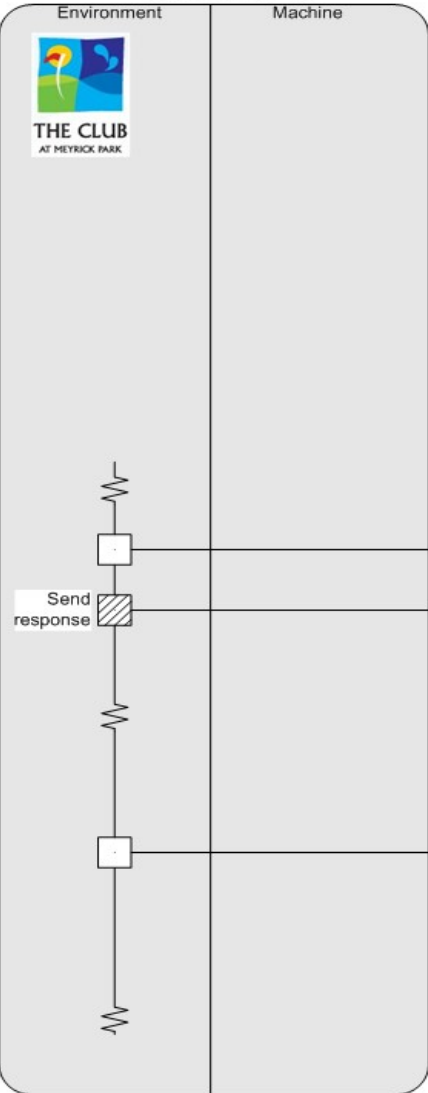
## Brighttime ✓ Environment



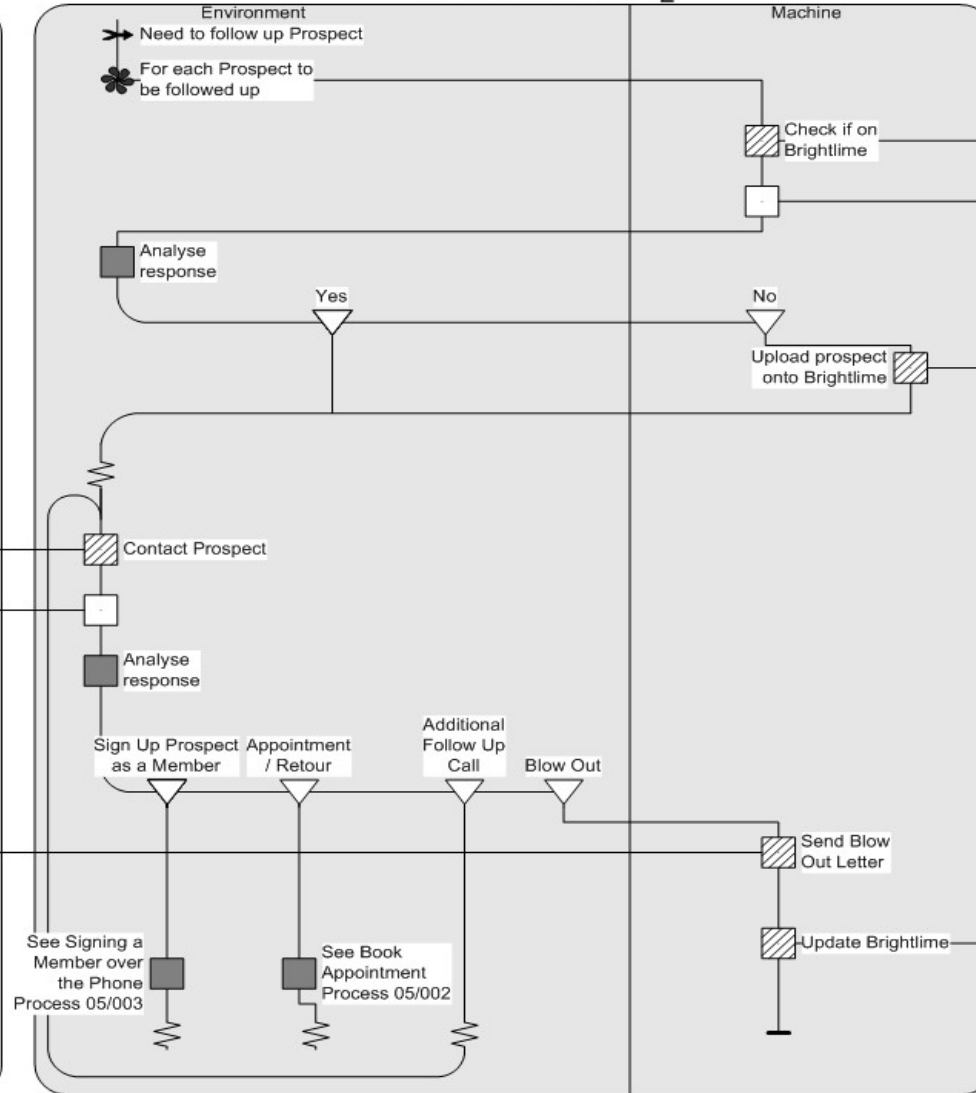


# Shared RAD

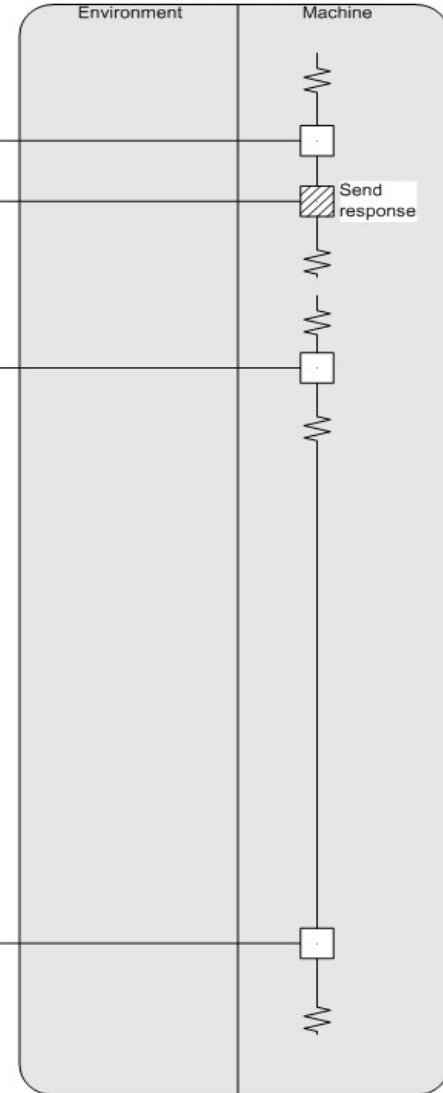
## Customer ✓



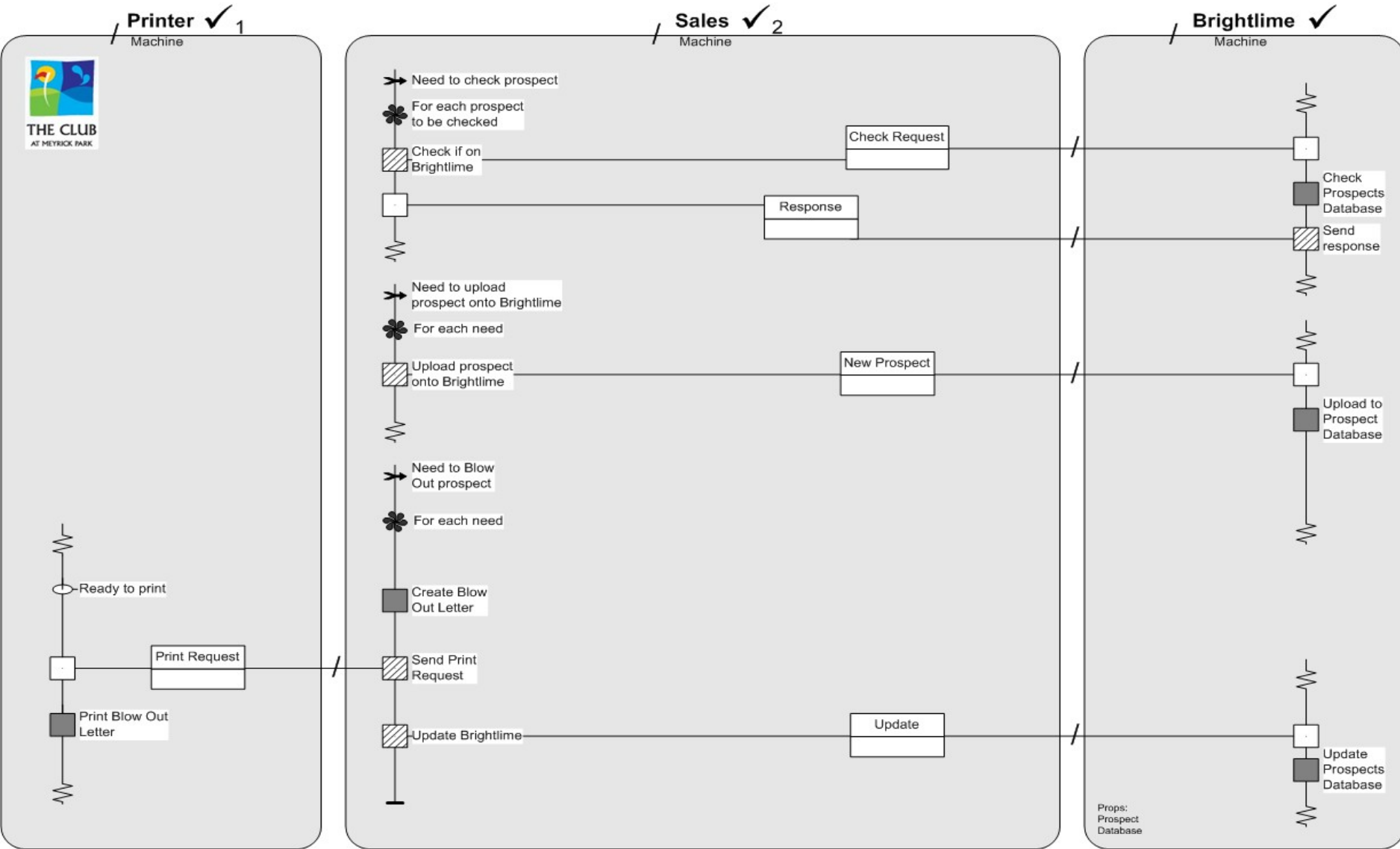
## Sales Advisor ✓<sub>2</sub>



## Brighttime ✓



# Machine RAD



- RAD (Transformation) and UML (Platform) Meta-models
- from
  - RADs (Machine)
- to
  - Class Diagram
  - Activity Diagram
  - Use Case



**Bournemouth  
University**

# Transformation - Initial Rules (1)

NATURAL ENGLISH	RAD	UML Class Diagram	UML Activity Diagram	UML Use Case	Example
Noun referring to human or system	Role	Class	Activity Partition	Actor; Relationship to Use Cases derived from that Role	Project Manager
Transitive verb with direct object (noun)	Independent Activity	Operation; Class & Association if Object results from Activity	Activity	Use Case; Chunk of Activity may define a single Use Case	Writes report
Clause where sequence is defined (before or after)	Looping, Line and Descriptor States	Attribute (only applicable for Descriptor States)	Transition; Synchronisation Bar if flow is split	Check context; May Define extend/include relationships	Ready to write report; Writes Report; Ready to send report; Sends Report
Clause joined with "or" Conjunction	Case Refinement (Alternatives)	Attribute	Decision Diamond; Guard	Relationship; Check context; May define extend/include relationships	Write report or Delegate Task
Clause joined with "and" conjunction	Part Refinement (Concurrency)	Attribute; Composition if refinement objects are descendent from resulting object	Synchronisation Bar	Include relationship	Writes and Sends Report; Write report a) and report b) to be contained in report c)
Sentence containing Role subject and object nouns with transitive verb; optionally modified by adverb	Interaction	Association; Operation in Role Classes. Aggregation if Role interactions are exclusive to only a single Role	Activity; Transition	Source and Destination Use Case; Relationship. Chunk of activity may define a single Use Case	Project Manager sends Report to General Manager and Contractor Quickly

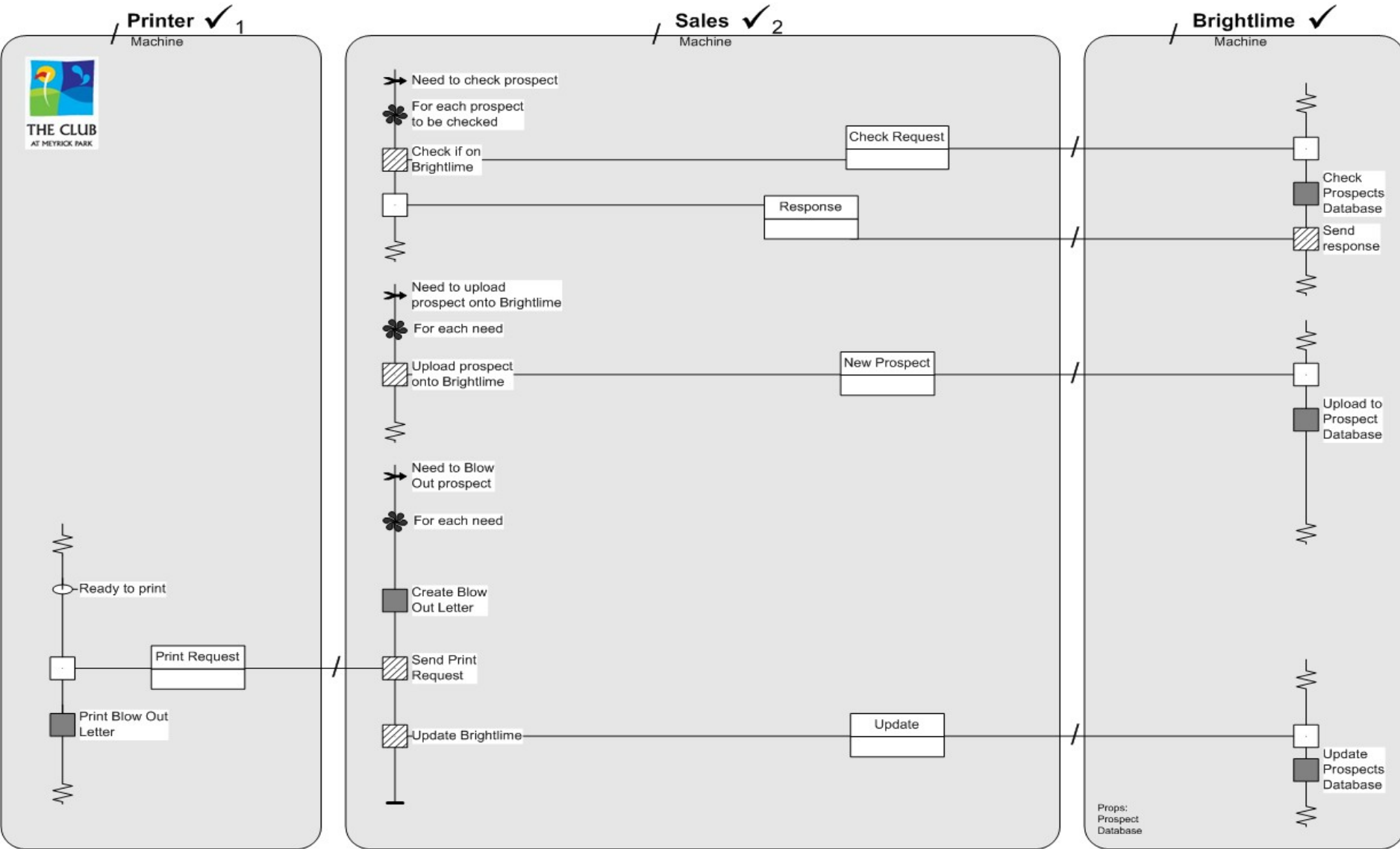


**Bournemouth  
University**

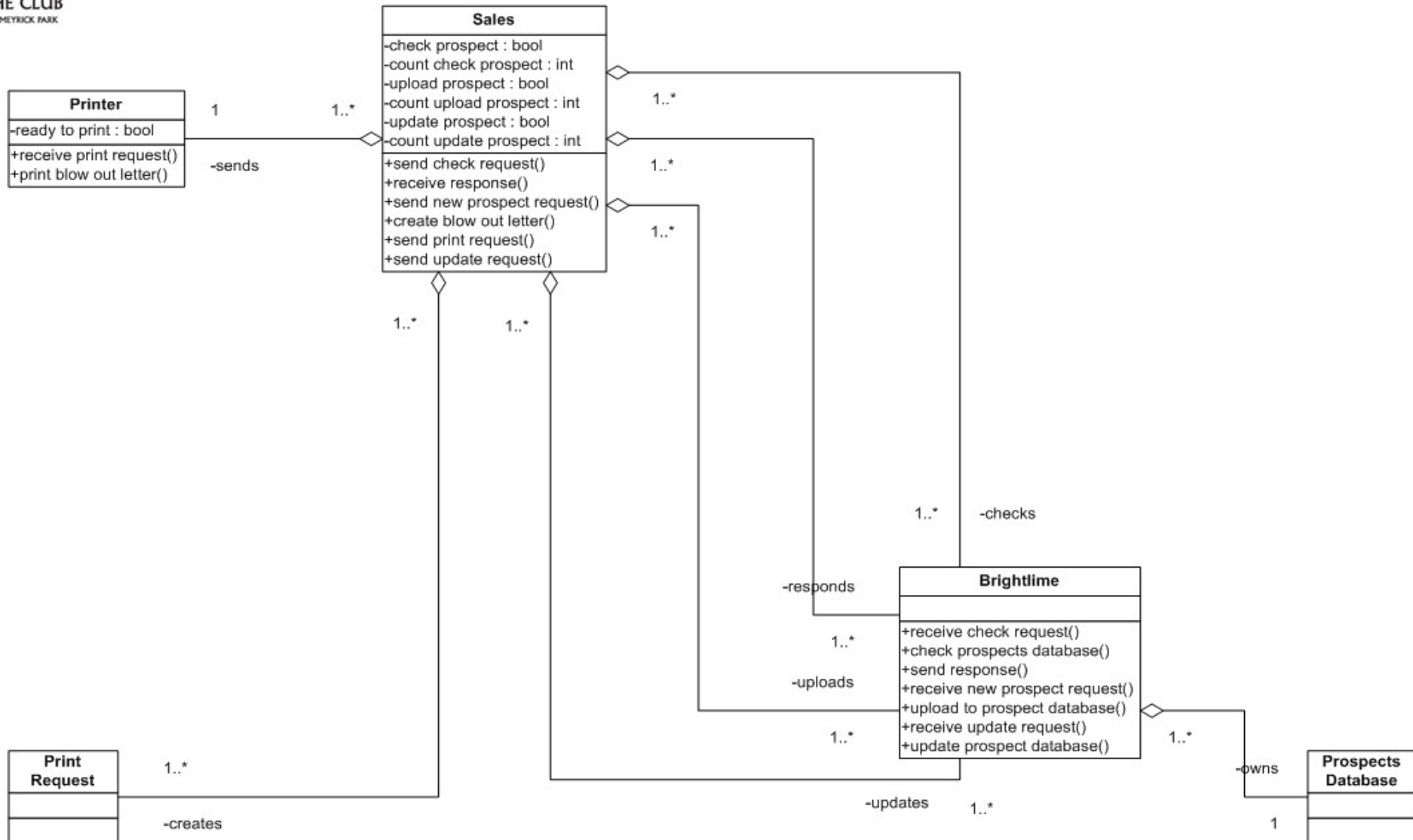
# Transformation - Initial Rules (2)

Transitive verb with Role noun initiating new Role noun	Start Role	Role Class; Operation in Source Role Class; and Association	Activity Partition; Activity and Transition	Actor	Project Manager selects Contractor
Noun referring to an event that starts a process	Trigger	Attribute	Start; Note	Note	Complaint is received
Quantifier associated with activity	Replication	Count attribute	Decision Diamond; Guard; Transition (loop) encapsulating replicated activity	Note	For every application received, assess it
Where sequence is undefined (before or after)	Undefined	Check context; May define alternate Class Diagram	Check context; May define alternate Activity Diagram; Transition; Stop	Check context; May define alternate Use Case; Relationship	The project manager writes report; The Project Manager owns a car
Determiner associated with Role noun	✓	Multiplicity	Note	Multiplicity	There are 500 employees; The Project Manager
Transitive verb and noun consumed by Role noun	Prop	Class; Association with Role Class	Note	Check context; May define alternate Actor; Use Case; Note	Uses database
Sequence terminating Transitive verb	Stop	Attribute in originating Role Class	Stop	Note	Project Ends
Adjective modifying noun or Pronoun	Note	Note	Note	Note	Project Manager is logged in

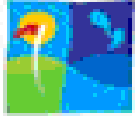
# Machine RAD



# Transformation – Class Diagram

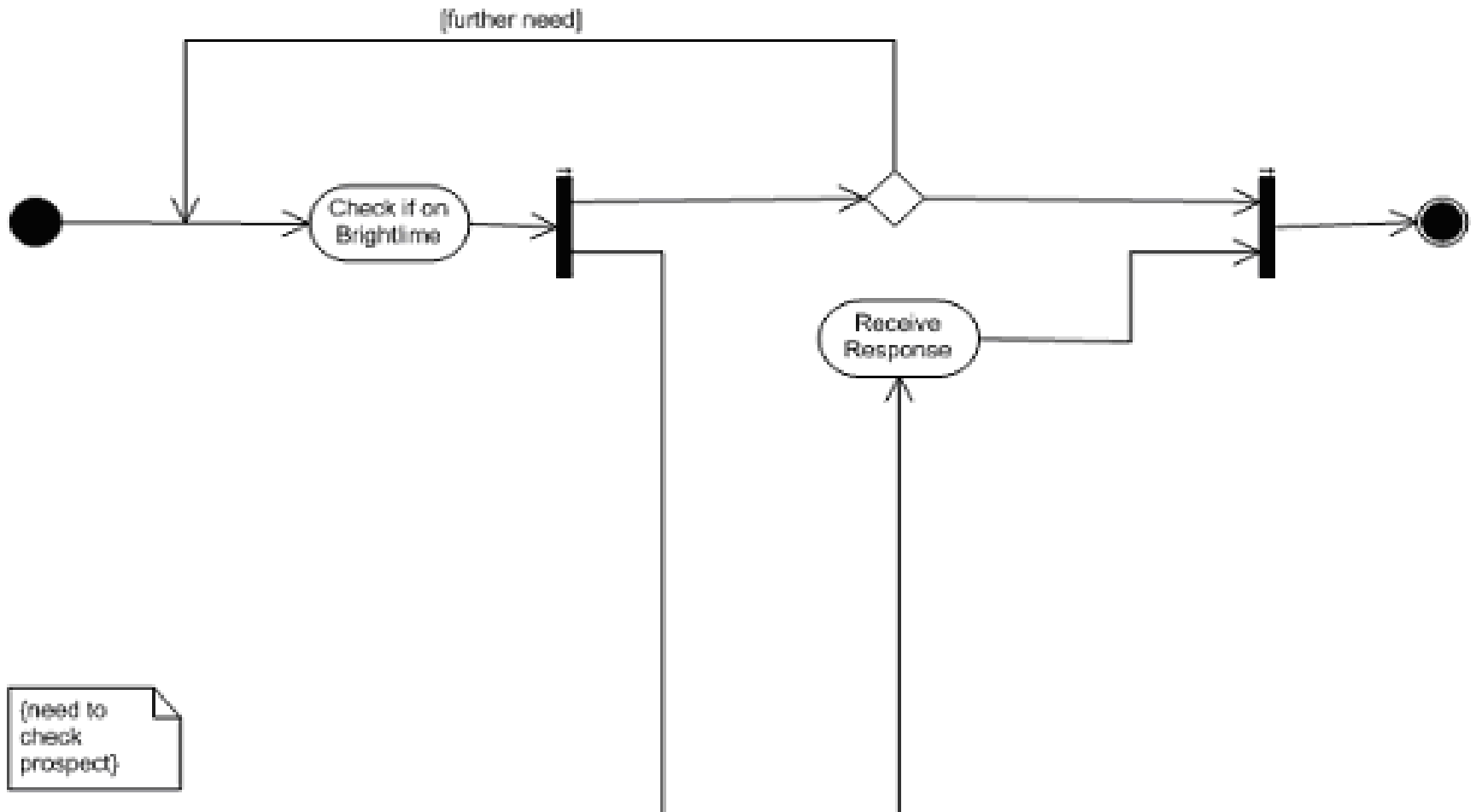


# Transformation – ‘Check Prospect’ AD (1)



THE CLUB  
AT THE TRACK INN

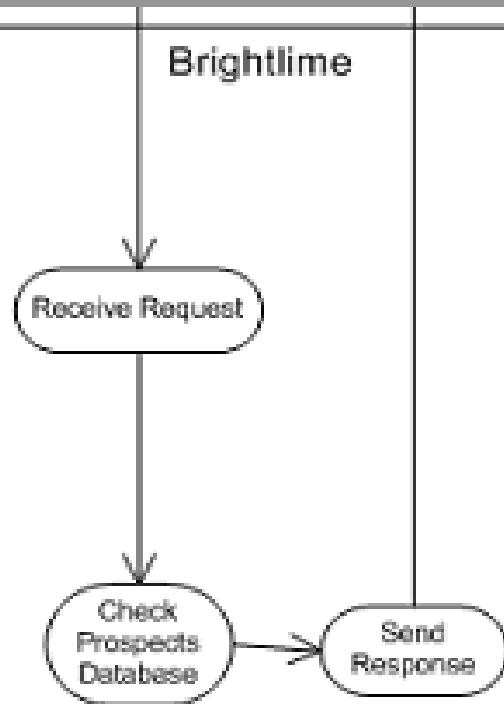
## Sales



(need to  
check  
prospect)



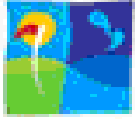
# Transformation – ‘Check Prospect’ AD (2)



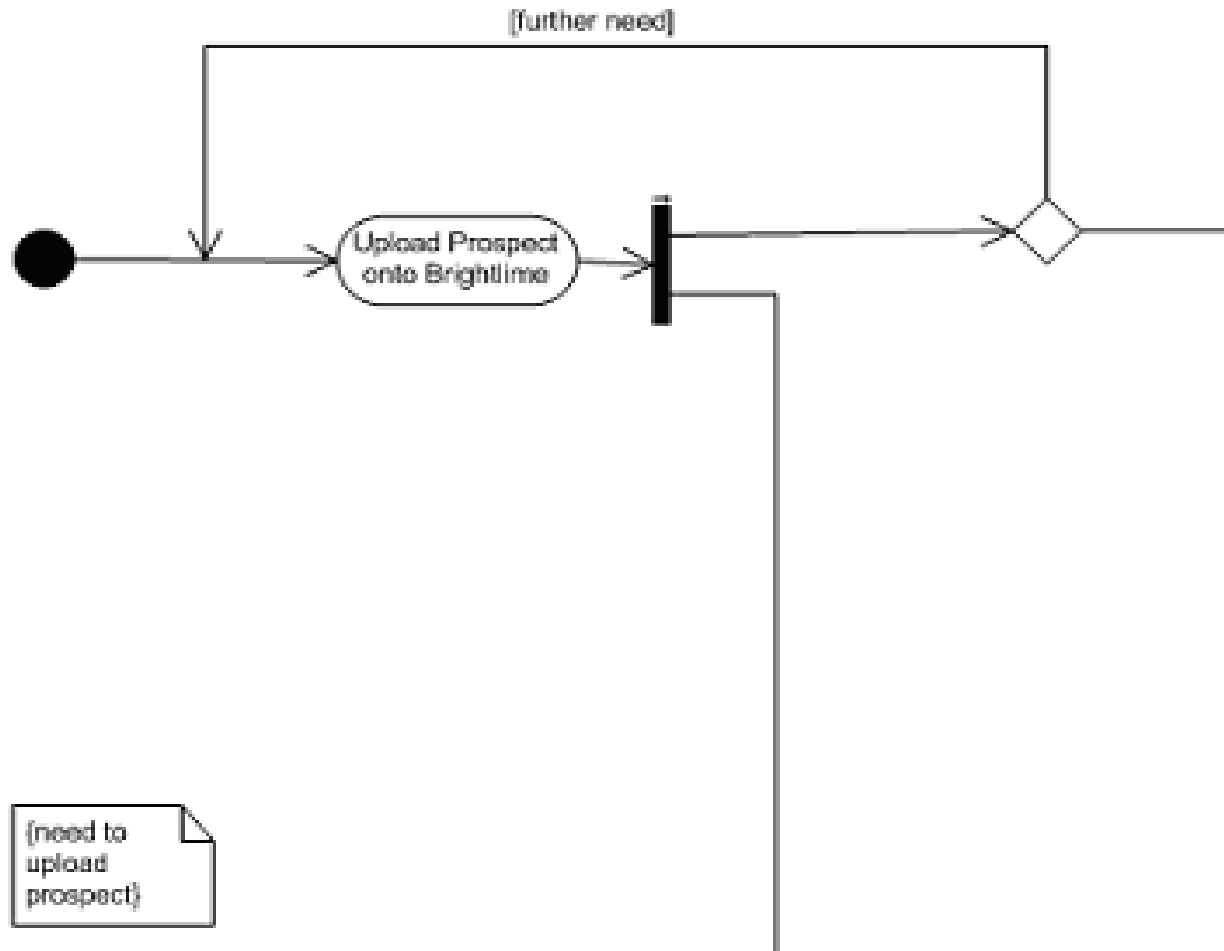
{prospect  
database  
exists}

# Transformation – ‘Upload Prospect’ AD (1)

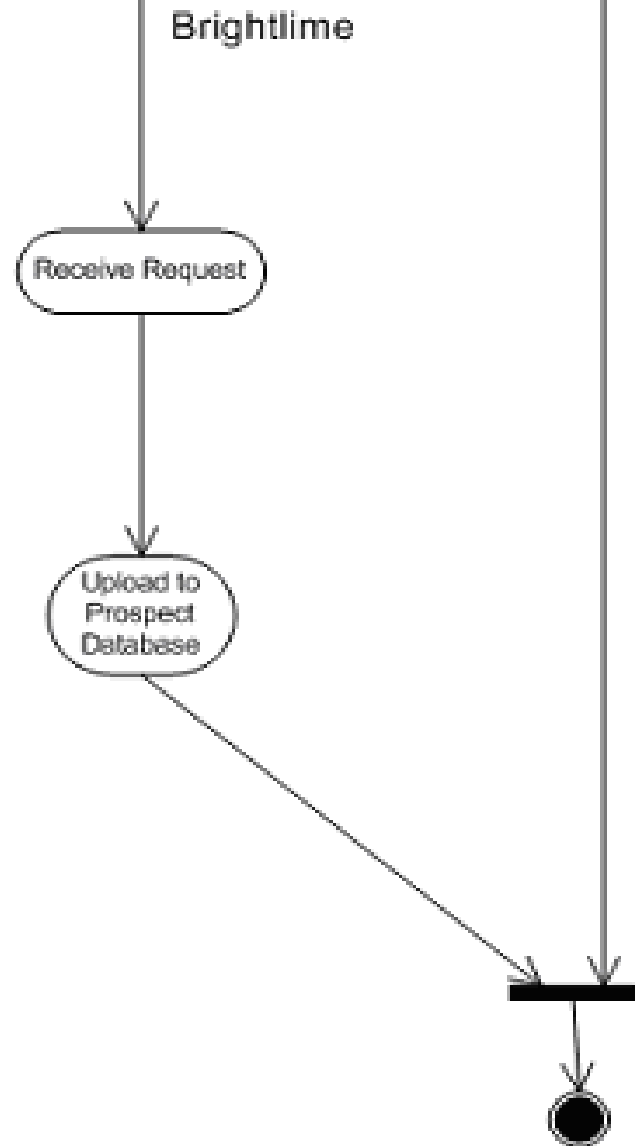
Sales



THE CLUB  
AT HETTRICK BARR

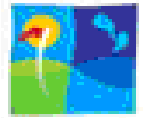


# Transformation – ‘Upload Prospect’ AD (2)

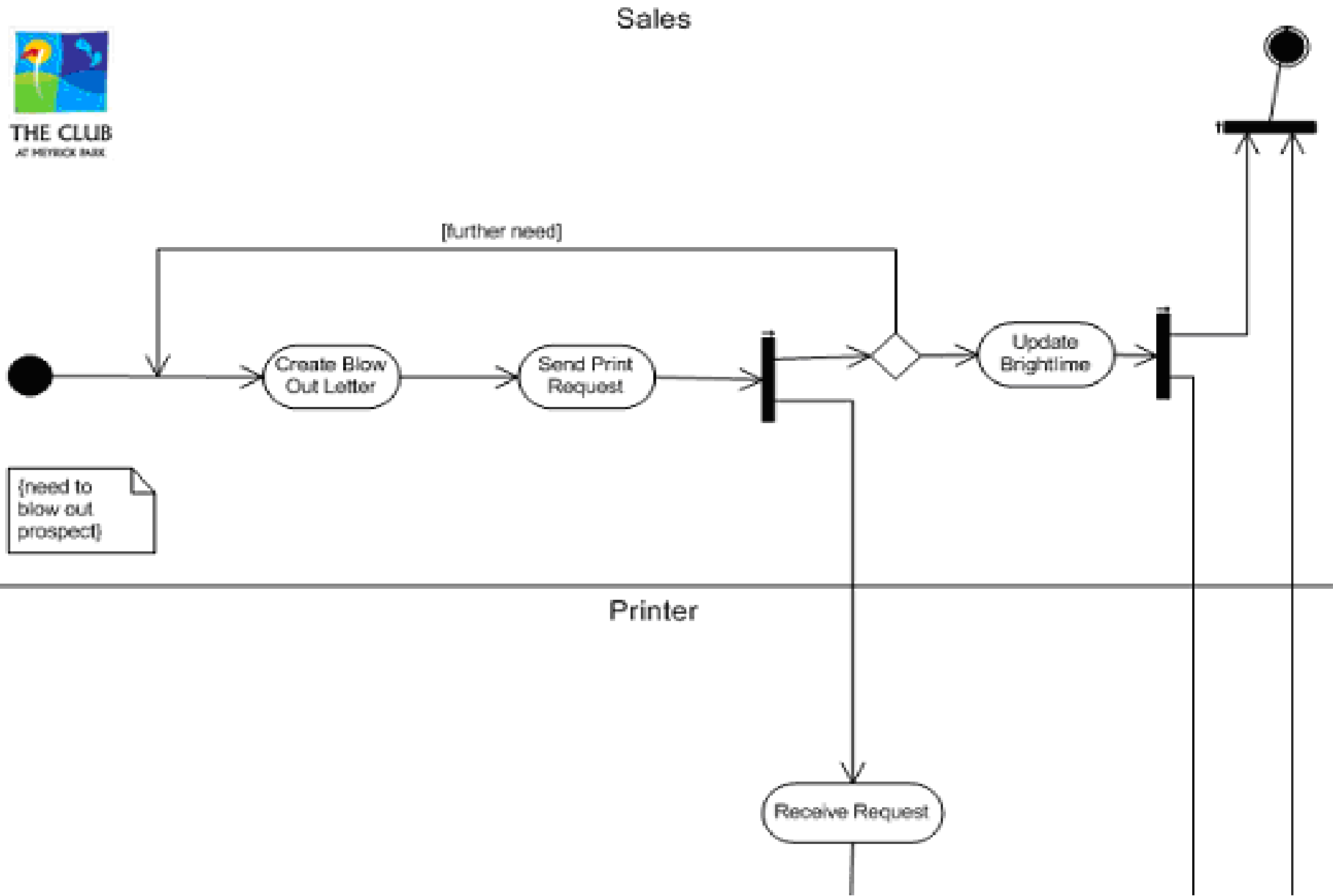


{prospect  
database  
exists}

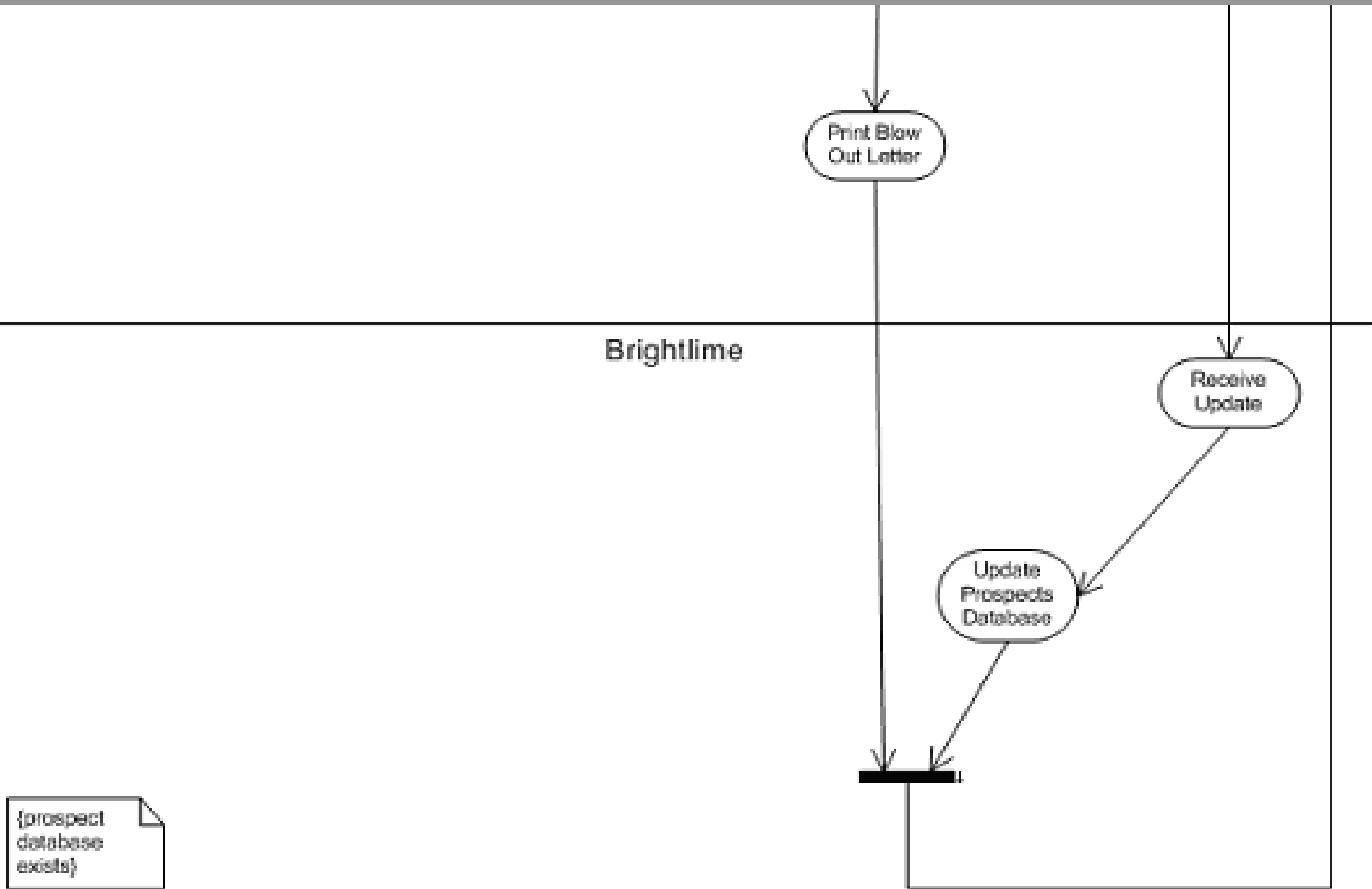
# Transformation – ‘Blow Out Prospect’ AD (1)



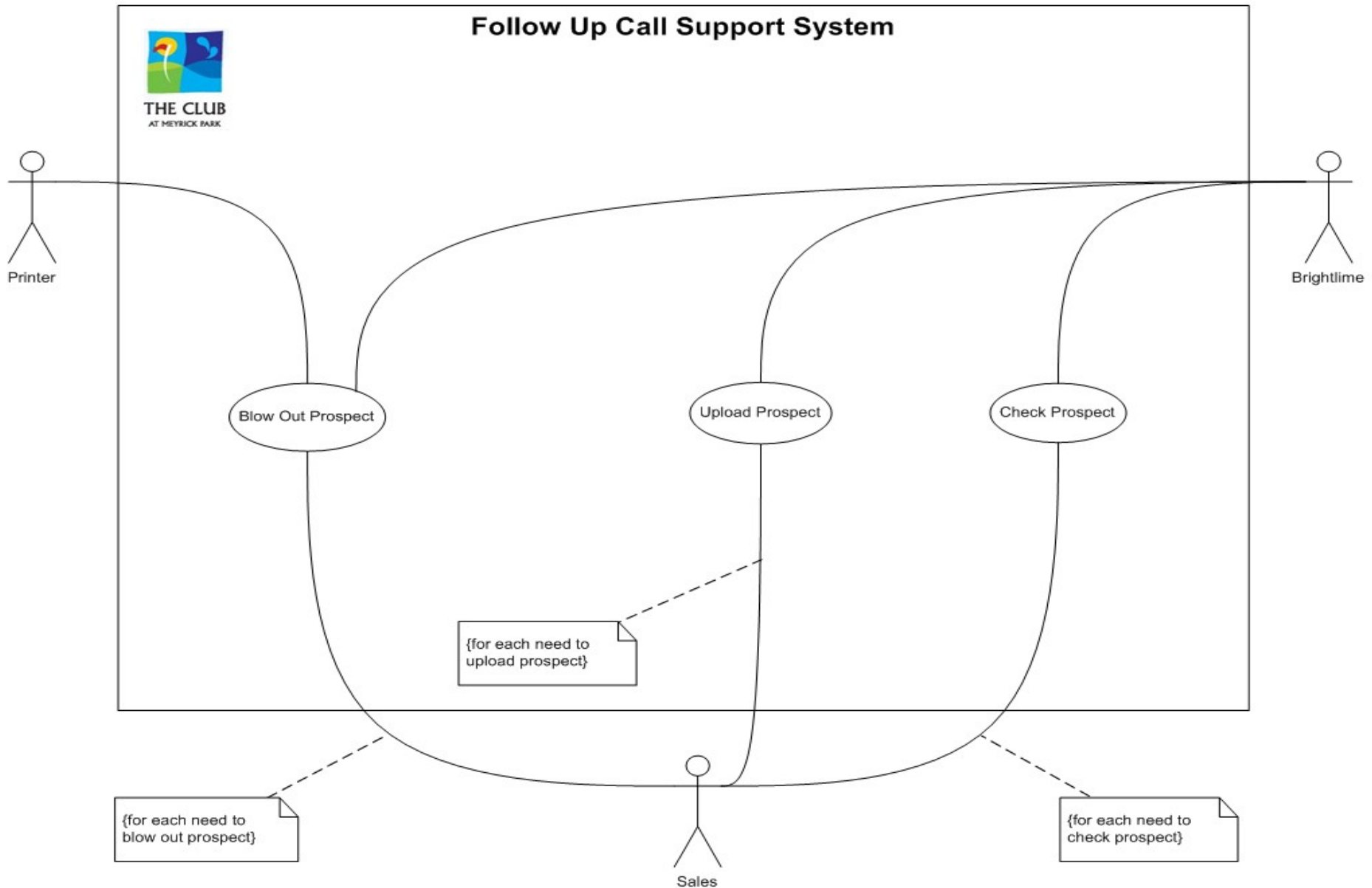
THE CLUB  
AT HENRICK BANK



# Transformation – ‘Blow Out Prospect’ AD (2)

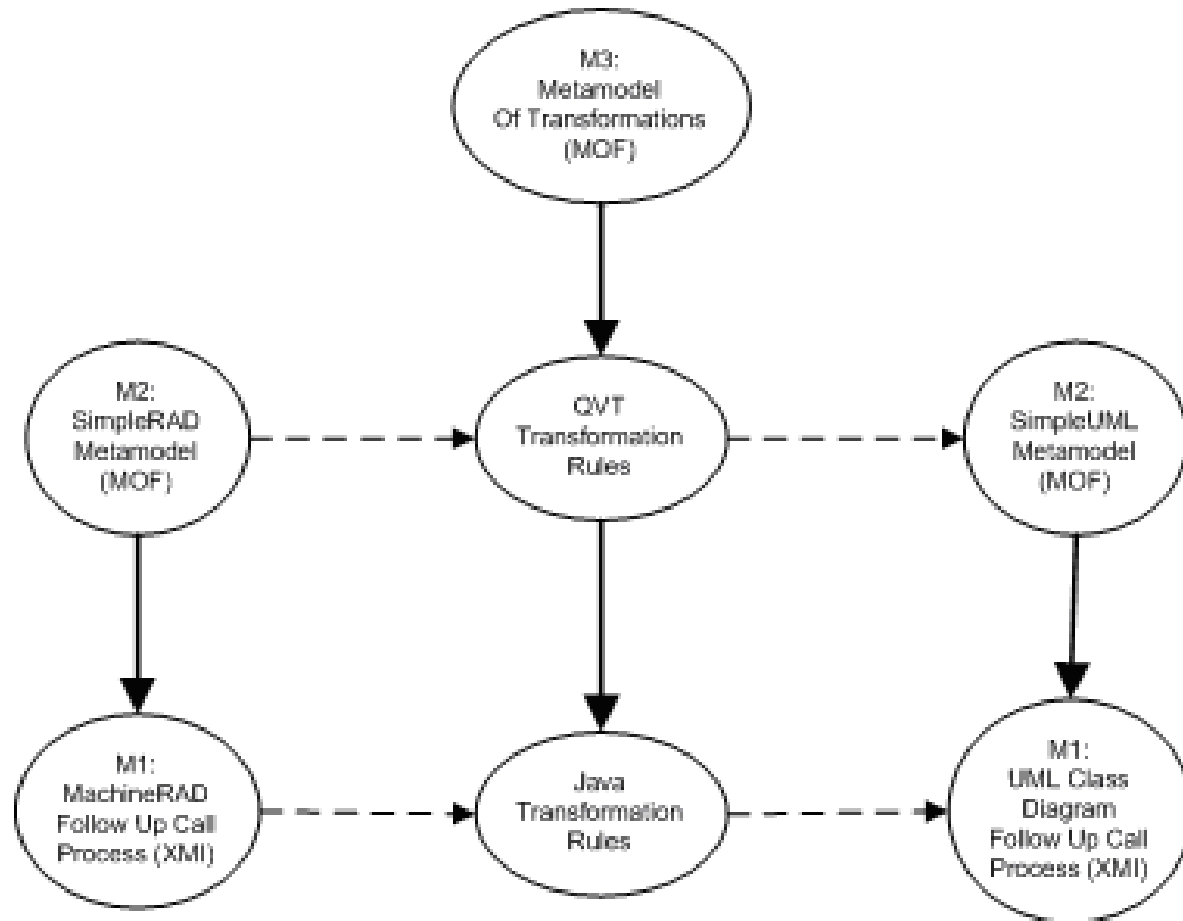


# Transformation – Use Case Diagram



- The declarative QVT-Relations (QVT-R) language forms part of the QVT standard central to the MDA and can be used to define transformation relations at the M2: Meta level between models conforming to the MOF meta-metamodel.
- *Role2Class* ( $r, c$ )
- *IndependentActivity2Operation* ( $ia, o$ )
- *Interaction2Operation* ( $i, o$ )
- *Prop2Class* ( $p, c$ ).

# QVT Transformation Pattern



*The extended rad2umlcd QVT transformation pattern*



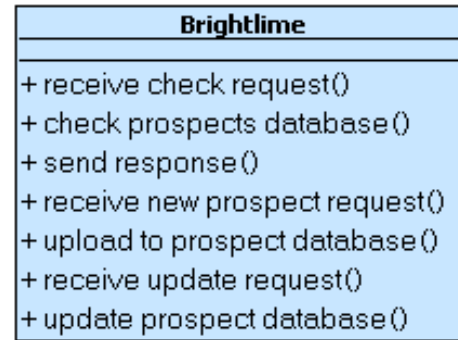
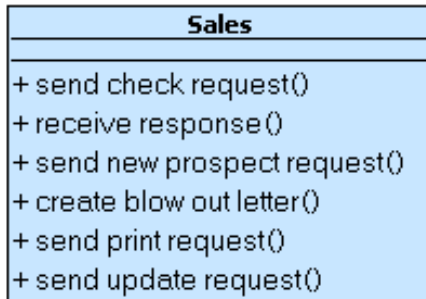
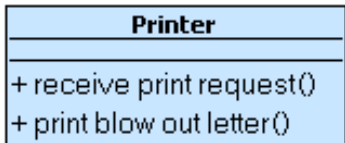


**Bournemouth  
University**

# Machine RAD VCLL XML

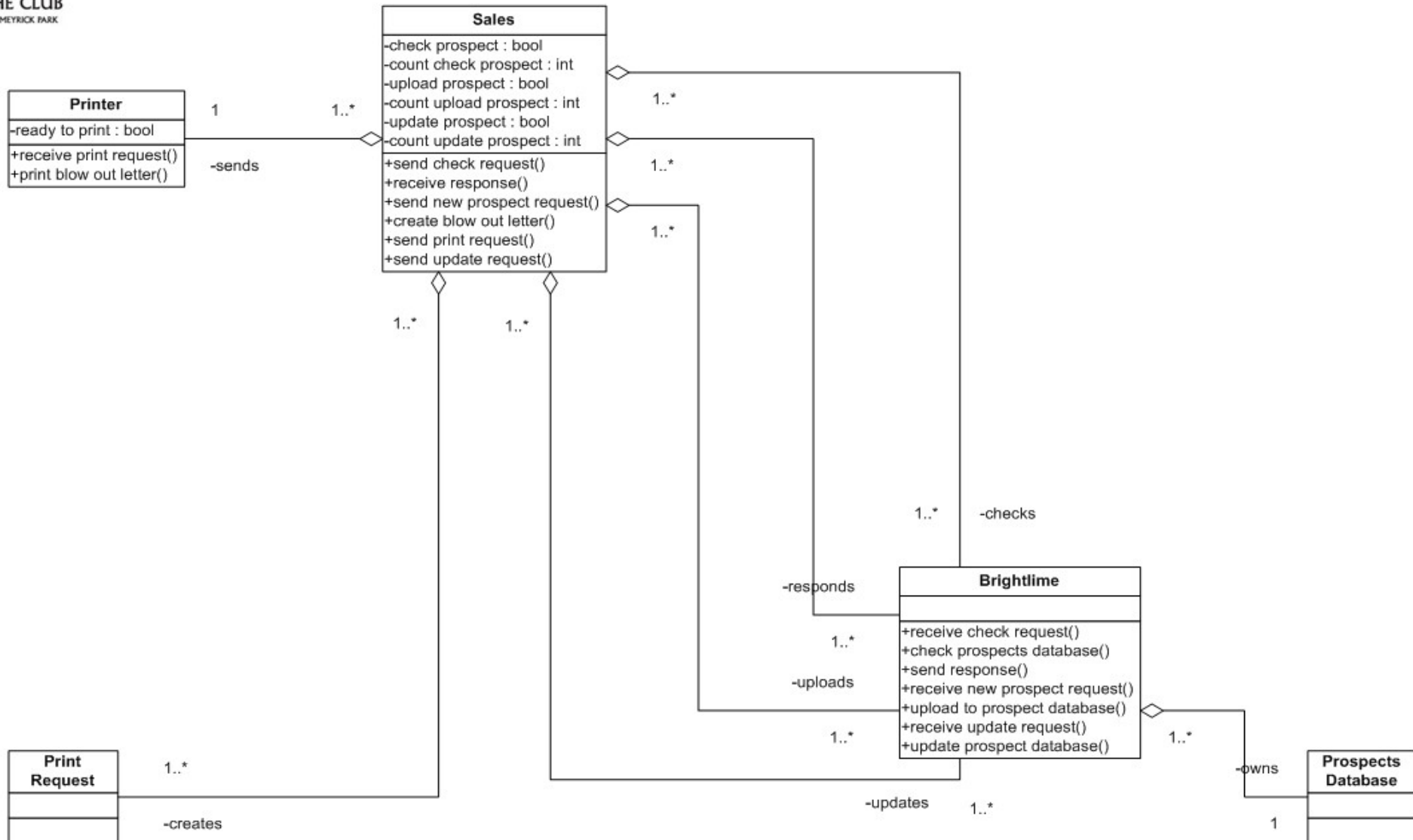
```
<?xml version="1.0" encoding="UTF-8"?>
<VclDiagram xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns="http://stp.eclipse.org/vcll" xmi:id="fusp" iD="fusp">
<pools xmi:type="Pool" xmi:id="p" iD="p" name="Printer">
    <vertices xmi:type="SubProcess" xmi:id="rpr" iD="rpr" outgoingEdges="1" incomingEdges="2" name="receive print request"/>
    <vertices xmi:type="SubProcess" xmi:id="pbol" iD="pbol" outgoingEdges="1" incomingEdges="2" name="print blow out letter"/>
</pools>
<pools xmi:type="Pool" xmi:id="s" iD="s" name="Sales">
    <vertices xmi:type="SubProcess" xmi:id="scr" iD="rcr" outgoingEdges="1" incomingEdges="2" name="send check request"/>
    <vertices xmi:type="SubProcess" xmi:id="rr" iD="rr" outgoingEdges="1" incomingEdges="2" name="receive response"/>
    <vertices xmi:type="SubProcess" xmi:id="snpr" iD="snpr" outgoingEdges="1" incomingEdges="2" name="send new prospect request"/>
    <vertices xmi:type="SubProcess" xmi:id="cbol" iD="cbol" outgoingEdges="1" incomingEdges="2" name="create blow out letter"/>
    <vertices xmi:type="SubProcess" xmi:id="spr" iD="spr" outgoingEdges="1" incomingEdges="2" name="send print request"/>
    <vertices xmi:type="SubProcess" xmi:id="sur" iD="sur" outgoingEdges="1" incomingEdges="2" name="send update request"/>
</pools>
<pools xmi:type="Pool" xmi:id="bl" iD="bl" name="Brightlime">
    <vertices xmi:type="SubProcess" xmi:id="rcr" iD="rcr" outgoingEdges="1" incomingEdges="2" name="receive check request"/>
    <vertices xmi:type="SubProcess" xmi:id="cpd" iD="cpd" outgoingEdges="1" incomingEdges="2" name="check prospects database"/>
    <vertices xmi:type="SubProcess" xmi:id="sr" iD="sr" outgoingEdges="1" incomingEdges="2" name="send response"/>
    <vertices xmi:type="SubProcess" xmi:id="rnpr" iD="rnpr" outgoingEdges="1" incomingEdges="2" name="receive new prospect request"/>
    <vertices xmi:type="SubProcess" xmi:id="utpd" iD="utpd" outgoingEdges="1" incomingEdges="2" name="upload to prospect database"/>
    <vertices xmi:type="SubProcess" xmi:id="rur" iD="rur" outgoingEdges="1" incomingEdges="2" name="receive update request"/>
    <vertices xmi:type="SubProcess" xmi:id="upd" iD="upd" outgoingEdges="1" incomingEdges="2" name="update prospect database"/>
</pools>
<pools xmi:type="Pool" xmi:id="pd" iD="pd" name="Prospects Database">
</pools>
</VclDiagram>
```

# Class Diagram (first-cut)

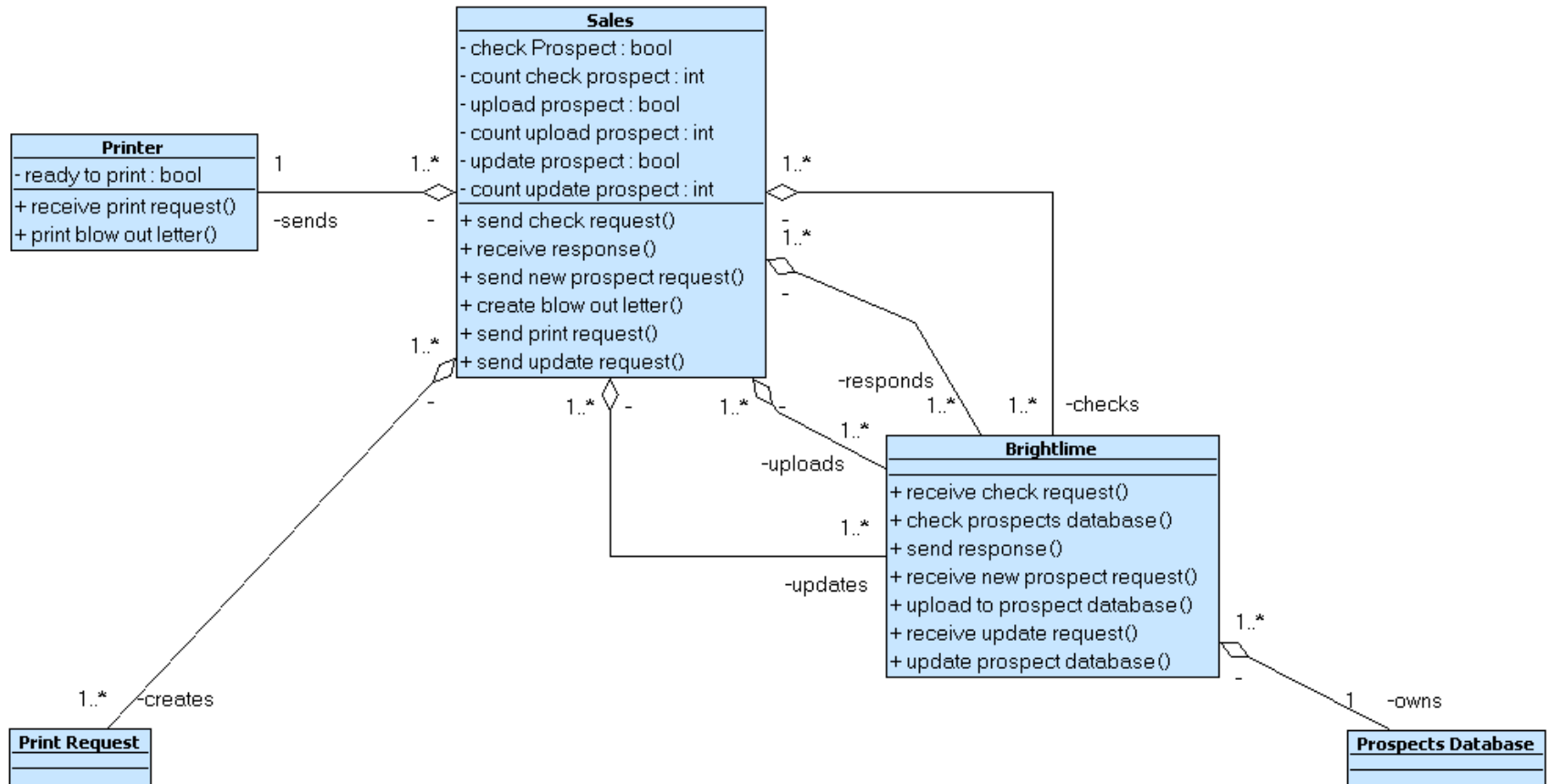


*Generated UML Class Diagram for the Follow Up Call process Machine RAD created from the application of QVT transformation rules in Java (first-cut)*

# Transformation – Class Diagram



# Class Diagram (modified)



Generated UML Class Diagram for the Follow Up Call process Machine RAD created from the application of QVT transformation rules in Java (modified)

# Questions

- Not an automatic process (semi-automatic with tool support) – Do we want it to be?
- Enterprise Systems – Can this really work?
- Systems Development – Do we want systems to be designed along natural business roles?

# Contributions

- The justification for and description of an extended framework into which different notations and tools can be placed to facilitate the accessibility of the MDA to business users;
- A unique method, including a mechanism for evolving an analysis RAD into a RAD suitable for specification within the MDA by extending the RAD notation, and rules to transform RAD elements into the UML (e.g. the RUD) derived from the *SimpleRAD* and *SimpleUML* metamodels (developed by this research), conforming to the MOF; and
- The verification of the extended framework and associated method, demonstrating the viability and accessibility of the xMDA to academia as learnable, and to industry as applicable, highlighting analysis problems often overlooked by the MDA when applied to commercial processes, and practical in terms of MDA tools and techniques.

# Conclusions

- By giving consideration to the tools and techniques available to RE and the MDA, it is possible to make the connection between domain and software models
  - MDA: OMG (2003) *MDA Guide* Version 1.0.1
    - [www.omg.org/mda/](http://www.omg.org/mda/)
  - RAD: Martyn A. Ould (2004) *Business Process Management: A Rigorous Approach*
    - <http://www.veniceconsulting.co.uk/>

Software Systems Research Centre  
Bournemouth University, Poole, Dorset, BH12 5BB.  
[AFouad@bournemouth.ac.uk](mailto:AFouad@bournemouth.ac.uk)