

# The Concept of an Adaptable Meta Model to Capture Dynamic System Behavior

Ashirul Mubin  
 The Graduate School  
 The University of Alabama  
 Tuscaloosa, AL 35487  
 amubin@ua.edu

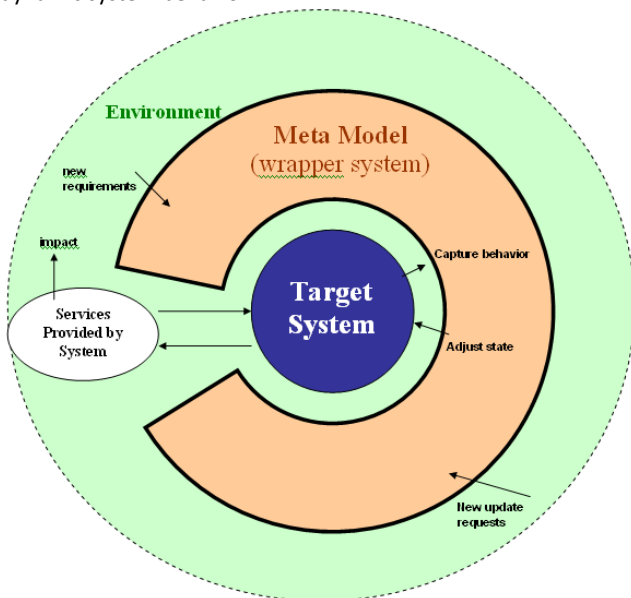
Daniel Ray  
 Department of Mathematics & Computer Science  
 The University of Virginia  
 Wise, VA 24293  
 dar5p@mcs.uvawise.edu

## Abstract:

While building a system, even with multiple development iterations, such as in Agile development strategies, beginning from the requirements analysis, design, developing, testing until its implementation and deployment, we meticulously take into account for the current and near-future needs of its business processes. However, often it is quite difficult to foresee further emerging demands that will eventually come up in the long run along with the continuous shifting of customer demands. At that point in the lifecycle, the value of the services provided by the system will begin to diminish at each encounter, because of its being unable to meet the newly emerged requirements. To effectively address such difficulties and avoid costly reengineering process in large systems, we need to capture the dynamic system behavior while the system is providing services, by gradually accumulating user activities in the workflow, its data utilization and changes in user satisfaction indexes at various probing points in the system. Based on these factors, we present a methodology for developing a meta-model that is integrated with the system itself. Such meta-model is adaptable to any forthcoming changes; and it propagates the changes and transforms into updated system state for incorporating new changes into the system when the service value degrades beyond a threshold level. By studying the track record of update request history and user activity log, we observe a significant gain in extended system lifecycle by applying such methodology over a period of several years.

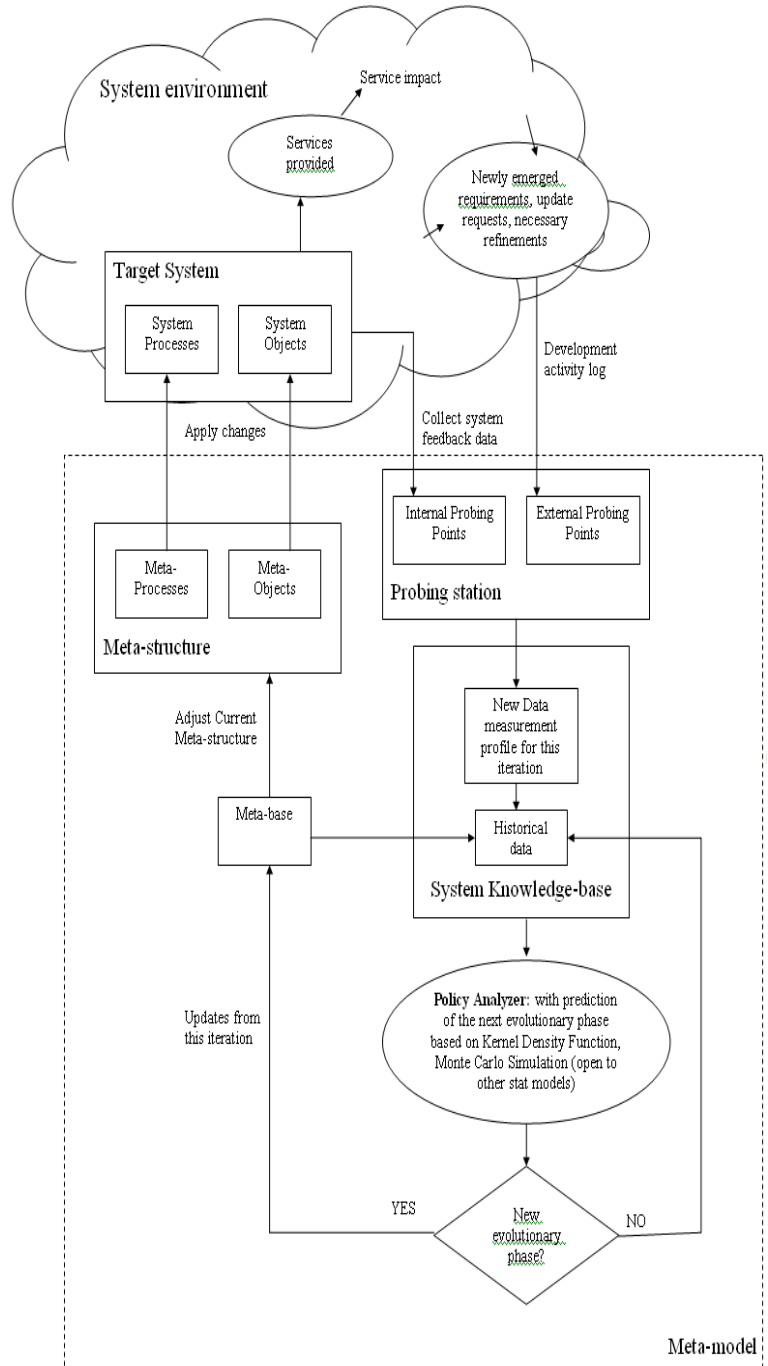
## Concept of Applying Meta-Model:

A meta-model can be seen as a wrapper system to capture dynamic system behavior.



## Iterative Architecture:

Three levels of iterations continue until system becomes steady upon each evolutionary phases.



(c) Copyright Ashirul Mubin 2009. This work is the intellectual property of the author. Permission is granted for this material to be shared for non-commercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of the author. To disseminate otherwise or to republish requires written permission from the author.