

# Architectural Analysis of Microsoft Dynamics NAV

[Awaiting approval for publication from Microsoft]

Tom Hvitved  
hvitved@diku.dk  
Department of Computer Science  
University of Copenhagen

November 18, 2008

## Abstract

This report describes our hands-on experience with the Enterprise Resource Planning (ERP) system Microsoft Dynamics NAV. Much literature exists on Microsoft Dynamics NAV, but none seem to have computer scientists as the main audience. We fill this gap by presenting the architecture of Microsoft Dynamics NAV using well-known concepts and terminology from computer science.

Our architecture analysis is *object based*, meaning that we present the “components” of NAV as classes in object oriented programming (OOP). During this analysis we address *upgradability* and *performance* problems. Our main observations are presented as *hypotheses*, the most important of which are (1) Lack of database *joins* – and in general database *views* – exacerbates an unnormalized database design which affects both performance and upgradability, (2) Lack of well-defined modules with well-defined interfaces affect upgradability negatively – introduction of joins is needed in order to redesign the software architecture in a modular fashion, and (3) Two key features of Microsoft Dynamics NAV, Sum Index Field Technology and reporting, have performance problems in the current implementation on Microsoft SQL Server.

We propose solutions for (1) - (3) which all take into account that Microsoft Dynamics NAV must be *backwards compatible*: The supply-chain of Microsoft Dynamics NAV is from Microsoft to customers via *partners*, who customize/modify the base product to meet the requirements of the customer. If customers are to upgrade to a new version of the ERP system, it is therefore crucial that the customizations can be ported as well – without having to implement the customizations from scratch. The proposed solutions for (3) are *asymptotically* faster than the current implementations.