

- Which issues about consistency this paper addresses?
 - Distribution of development implies distribution of models
 - Consistency is, by its definition, a check across views/perspectives/even models (=data)
 - If data is distributed then there exists the problem that local consistency checking is no longer feasible (lack of information)
 - Authors propose an approach to replicate the missing data in the locations where necessary to there perform consistency checking locally
 - Distribution problem is severe. This approach attempts a fully heterogeneous way of dealing with this problem

- What is the main novel contribution/approach of the paper?
 - Two ways of achieving this notion of distributed consistency checking
 - Continue at the other machine where the data is
 - Fetch data to local machine
 - What is actually better? This paper chose the latter
 - Data replication
 - Need for “jump points” to make replication incremental

- Do you see any problem with the approach/contribution of the paper?
 - Filters are used to understand which other data is needed for consistency checking -> impact matrix is bound to types
 - Data replication likely increases with every “jump” – assumption: without matrix, 5+ jumps will reach much of the entire model
 - Who gets to benefit from a consistency message (all views where at least one piece of data is used? If yes then propagation of inconsistencies is needed because evaluation starts at one view but may affect others
 - What if a piece of data was modified by others? Which version are you going to fetch?

- What would you suggest to improve the paper's contribution?
 - Validation: look at the scalability issue
 - Must maintain an editing sequence?
 - Look at approach that maintain exact and complete scope knowledge (xlinkit, model/analyzer) -