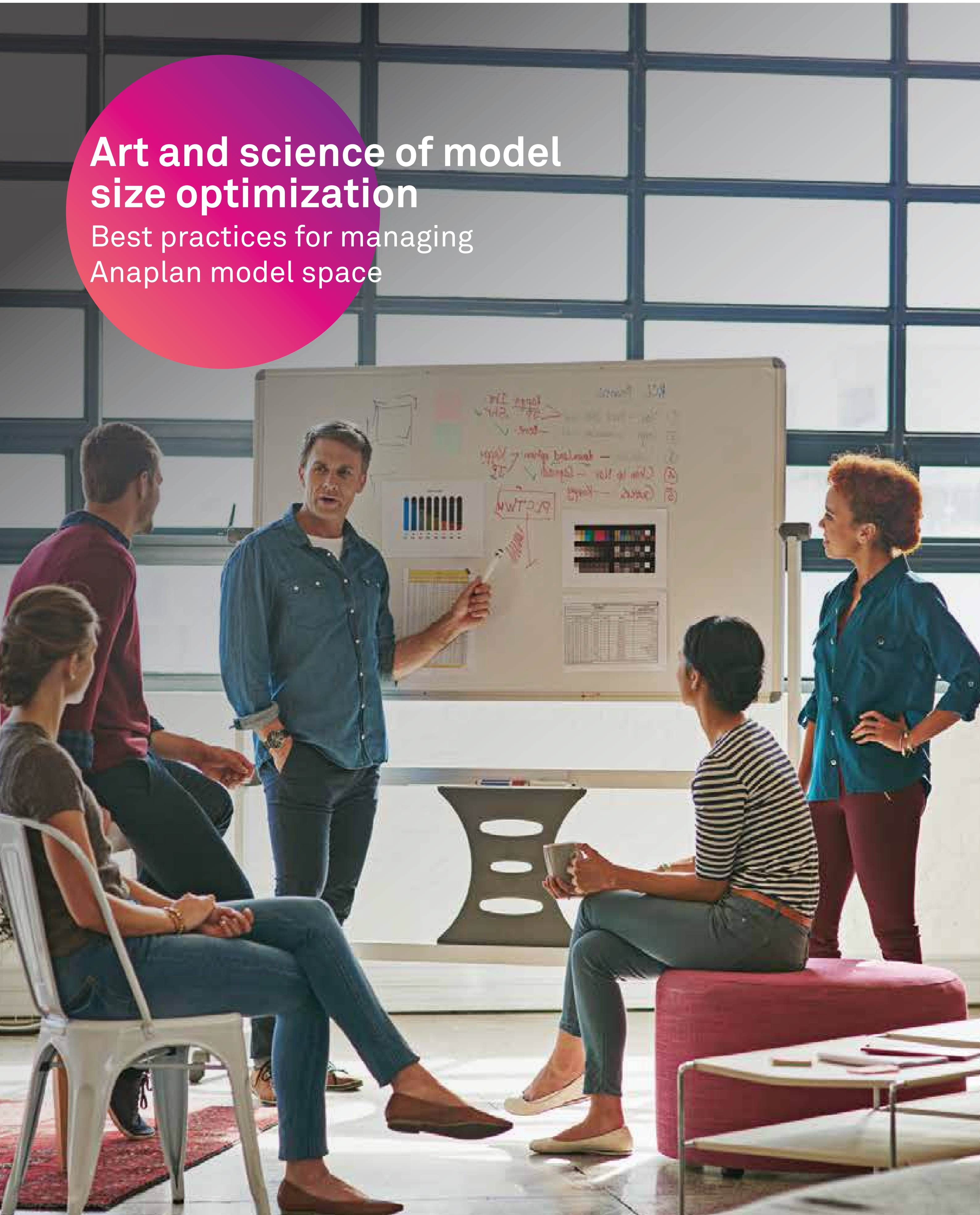
Art and science of model size optimization

Best practices for managing Anaplan model space



Every business needs to have a scalable, reliable model in place that addresses all business functionalities and enables understanding of business performance in real time. With business processes going agile, and the exact ask being seldom clear, it is difficult for the solution architect to size the business model accurately for addressing the requirements.

Anaplan connected planning platform, a Cloud-native solution that supports detailed models, delivers the potential to achieve success in this area. Anaplan models offer desirable levels of module capacity and flexibility to address business needs. Though Anaplan allows high levels of capacity, it is crucial to ensure optimized size while designing models. Excessive model size costs more and involves overheads like data duplication.

Anaplan flexibility can enhance the model, but at the same time, can give a tough time to the architect or model builder in terms of controlling the right size for the model. Hence, it is imperative to follow certain methods meticulously to bring model to the right size that meets the requirements. Points discussed below, followed in chronological order, will drastically reduce the model size and ensure that model builders have followed the most optimized approach to designing and building models.

1. Check on the time settings

Native time dimension settings works best in most of the cases, except for a few. For instance, if the business would like to view data weekly but compute the month totals based on days and not on weeks, Anaplan has no time settings to address this scenario. Hence, either a custom or a fake time scale has to be created with custom mapping of days to month separately in module to use model calendar settings. Another possible solution is to use Week General Settings but this would not allow certain functions such as Month Value. Best approach in this scenario is to use 4-4-5, 4-5-4 or 5-4-4 time distribution.

2. Take the POC approach

Proof of Concept approach should be followed as part of design or initial model building to cover the basic functionalities. For e.g. to prove which time dimension settings would work fine, best approach is to do it. Certain pertinent questions that should be answered are:

- What value additional days to month level aggregation would bring that week to month aggregation cannot bring?
- How important is the differential data for the decision making?
- Perform a POC and showcase the aggregation of the numbers from days to month and from weeks to month via small proof of concept. This will tear the facts apart and will provide insight regarding the data.

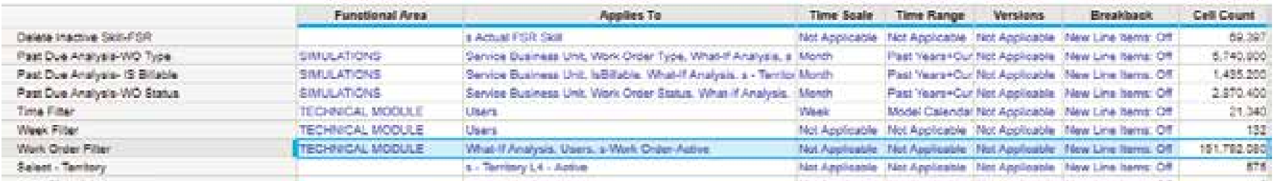
3. Evaluate usage of custom lists for versions

Use native versions wherever possible. However, you cannot create subsets against them. There can be certain modules whose dimensionality depends on What-if versions and not actual version. Idea is to create a subset against the custom list of versions. In Figure 1, a subset named s-simulation is created to limit the weekly module size to 151,792,040 Cells (1.43 GB). Actual version being included would increase model size.

4. Go for hierarchy pruning

There are many instances wherein unnecessary transactional level details are brought into Anaplan. Business should be challenged with questions regarding the relevant list members to be included in Anaplan. If there is a requirement to drill back to the transaction data from the model, then there is a valid reason to hold transactional data.

For instance, a company reduced the number of service business units from 19 to 5, leading to 70% space reduction. They were pruned as they were not needed in the Anaplan model. In Anaplan, all the modules will not contain all the



	Functional Area	Applies To	Time Scale	Time Range	Versions	Breakback	Cell Count
Delete Inactive Skill-FSR		s Actual FSR Skill	Not Applicable	Not Applicable	Not Applicable	New Line Items: Off	59,397
Past Due Analysis-WO Type	SIMULATIONS	Service Business Unit, Work Order Type, What-if Analysis, s	Month	Past Years+Cur	Not Applicable	New Line Items: Off	5,740,800
Past Due Analysis- IS Billable	SIMULATIONS	Service Business Unit, IsBillable, What-if Analysis, s - Territor	Month	Past Years+Cur	Not Applicable	New Line Items: Off	1,435,200
Past Due Analysis-WO Status	SIMULATIONS	Service Business Unit, Work Order Status, What-if Analysis,	Month	Past Years+Cur	Not Applicable	New Line Items: Off	2,870,400
Time Filter	TECHNICAL MODULE	Users	Week	Model Calendar	Not Applicable	New Line Items: Off	21,340
Week Filter	TECHNICAL MODULE	Users	Not Applicable	Not Applicable	Not Applicable	New Line Items: Off	132
Work Order Filter	TECHNICAL MODULE	What-if Analysis, Users, s-Work Order-Active	Not Applicable	Not Applicable	Not Applicable	New Line Items: Off	151,792,080
Select - Territory		s - Territory L4 - Active	Not Applicable	Not Applicable	Not Applicable	New Line Items: Off	578

Figure 1

lists, it is a careful selection of the lists that would determine the size of the module. The intent should be to bring data that is absolutely relevant and useful rather than copying from the source.

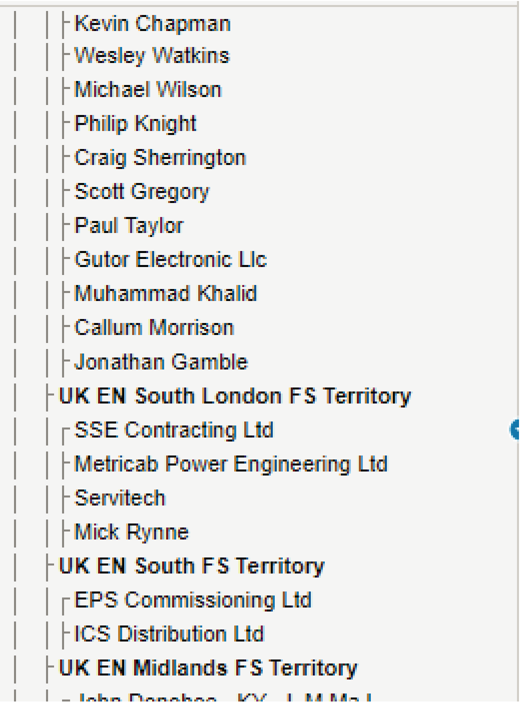
5. Do not cross big lists – Manage dimensionality

There can be instances where numbered lists which have 200K+ members are required to be built. Anaplan is capable of handling such huge lists, but there could be an issue when the lists crosses other bigger lists for e.g. weekly view for two years (53*2 = 106 Members). This can cause module size to increase.

The approach should be to use numbered list with limited line items. This will bring in control. Identify independent numbered lists to use in separate module to reduce the model size.

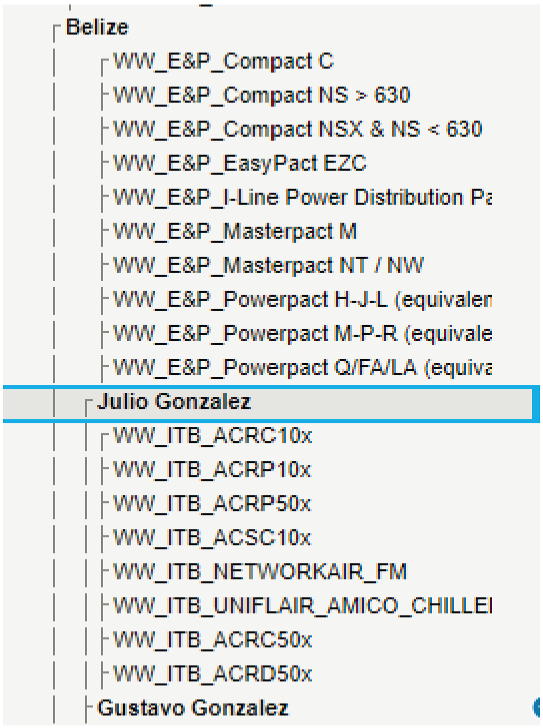
6. Use numbered lists

Usage of normal lists should be encouraged as maintenance issues are less and providing selective access is easier. In many cases, performance and maintenance is much better if you allow sparsity. However, there are situations where user needs control in terms of adding members to the lists. Numbered lists are one of the ways to control sparsity. It also provides flexibility to roll up against the different parents and create virtually a shared instance of the members. Creating multiple levels within numbered lists will help reduce sparsity. For e.g. Country->#Territory->#Sales Rep can help to map the right sales rep against right territory. Maintenance around numbered list should also be developed to provide foolproof system to the users.



```
│ │ ├Kevin Chapman
│ │ ├Wesley Watkins
│ │ ├Michael Wilson
│ │ ├Philip Knight
│ │ ├Craig Sherrington
│ │ ├Scott Gregory
│ │ ├Paul Taylor
│ │ ├Gutor Electronic Llc
│ │ ├Muhammad Khalid
│ │ ├Callum Morrison
│ │ ├Jonathan Gamble
│ ├UK EN South London FS Territory
│ │ ┌SSE Contracting Ltd
│ │ ├Metricab Power Engineering Ltd
│ │ ├Servitech
│ │ ├Mick Rynne
│ ├UK EN South FS Territory
│ │ ┌EPS Commissioning Ltd
│ │ ├ICS Distribution Ltd
│ ├UK EN Midlands FS Territory
```

Figure2a #Territories->#Sales Rep



```
┌Belize
│  ┌WW_E&P_Compact C
│  ├WW_E&P_Compact NS > 630
│  ├WW_E&P_Compact NSX & NS < 630
│  ├WW_E&P_EasyPact EZC
│  ├WW_E&P_I-Line Power Distribution Pa
│  ├WW_E&P_Masterpact M
│  ├WW_E&P_Masterpact NT / NW
│  ├WW_E&P_Powerpact H-J-L (equivalen
│  ├WW_E&P_Powerpact M-P-R (equivale
│  ├WW_E&P_Powerpact Q/FA/LA (equiva
│ ┌Julio Gonzalez
│ │ ┌WW_ITB_ACRC10x
│ │ ├WW_ITB_ACRP10x
│ │ ├WW_ITB_ACRP50x
│ │ ├WW_ITB_ACSC10x
│ │ ├WW_ITB_NETWORKAIR_FM
│ │ ├WW_ITB_UNIFLAIR_AMICO_CHILLEI
│ │ ├WW_ITB_ACRC50x
│ │ ├WW_ITB_ACRD50x
│ ├Gustavo Gonzalez
```

Figure2b #Sales Rep->#Skills

7. Use numbered lists in combination

Above example states that not all sales reps belong to all territories. A combination list created between #Territory and #Sales Rep that is dynamically maintained using Actions will reduce sparsity. Dynamic hierarchy management of two numbered lists as combination of lists can be time consuming to implement in some cases but worth implementing in addressing changing needs.

8. Use subsidiary views in a controlled manner

Ideally, modules with same dimensionality should be kept separate but there are such scenarios in any engagement where not all the line items in a module follow the same dimensionality, but you are bound to keep them in the same module due to functional reasons. Subsidiary views should be created by changing any one of the following:-

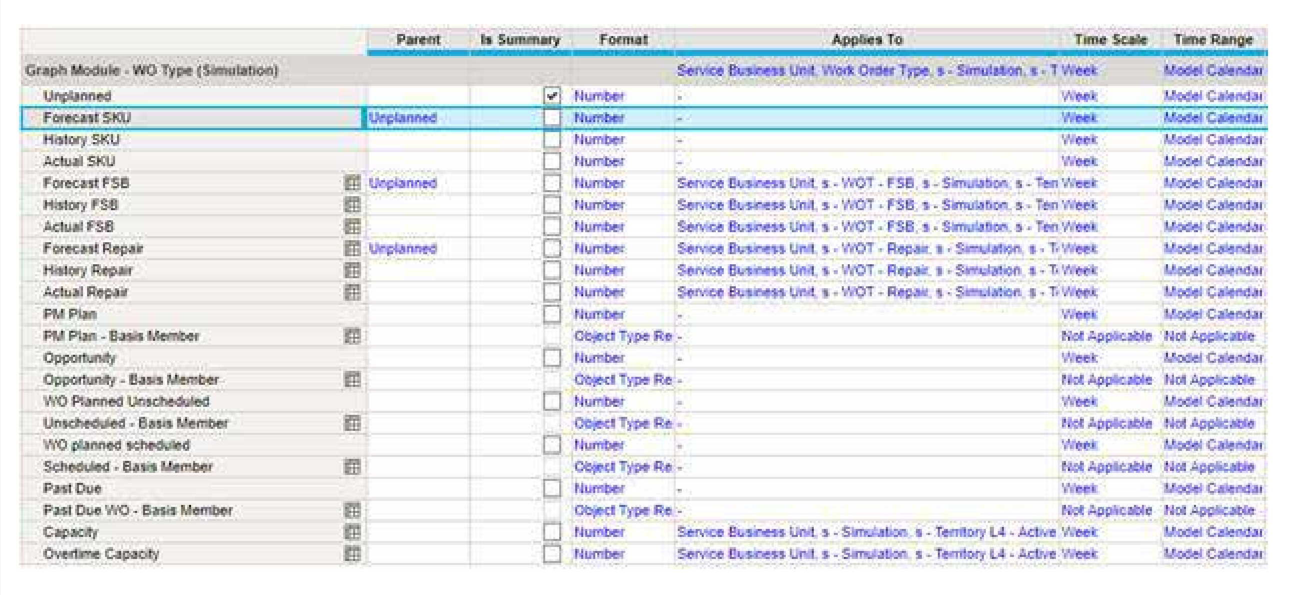
- Applies To
- Time Scale
- Time Range

Generally, if the module is a weekly module, monthly level of data has to be captured in line item leading to change in the Time scale for a line item and hence subsidiary view. Imagine the impact of reducing the cell count to 12 from 53 (Month to Week) leading to close to 80% cell count reduction. If the relevant lists are kept against relevant line items, this can create wonders in terms of size reduction. Careful visit to all the modules should be made at the development stage itself.

9. Quick check on line items

Few quick checks on writing complex formula in single line item should be encouraged when working with larger modules. More line items = More cell count. Check the following:-

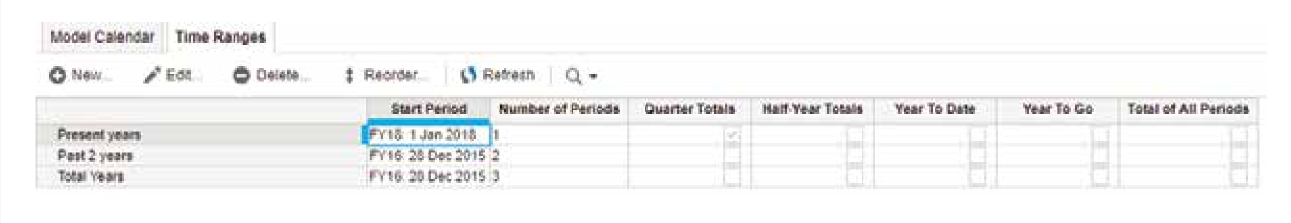
- Use 'No data' where possible
- Merge line items into single line item if possible by combining formulas, still maintaining readability and traceability
- Bring only relevant dimension values as line item that are key to aggregation for SUM function or LOOKUP function
- Delete unnecessary line-items



	Parent	Is Summary	Format	Applies To	Time Scale	Time Range
Graph Module - WO Type (Simulation)				Service Business Unit, Work Order Type, s - Simulation, s - T	Week	Model Calendar
Unplanned		☑	Number	-	Week	Model Calendar
Forecast SKU	Unplanned	☐	Number	-	Week	Model Calendar
History SKU		☐	Number	-	Week	Model Calendar
Actual SKU		☐	Number	-	Week	Model Calendar
Forecast FSB	Unplanned	☐	Number	Service Business Unit, s - WOT - FSB, s - Simulation, s - Ten	Week	Model Calendar
History FSB		☐	Number	Service Business Unit, s - WOT - FSB, s - Simulation, s - Ten	Week	Model Calendar
Actual FSB		☐	Number	Service Business Unit, s - WOT - FSB, s - Simulation, s - Ten	Week	Model Calendar
Forecast Repair	Unplanned	☐	Number	Service Business Unit, s - WOT - Repair, s - Simulation, s - T	Week	Model Calendar
History Repair		☐	Number	Service Business Unit, s - WOT - Repair, s - Simulation, s - T	Week	Model Calendar
Actual Repair		☐	Number	Service Business Unit, s - WOT - Repair, s - Simulation, s - T	Week	Model Calendar
PM Plan		☐	Number	-	Week	Model Calendar
PM Plan - Basis Member			Object Type Re	-	Not Applicable	Not Applicable
Opportunity		☐	Number	-	Week	Model Calendar
Opportunity - Basis Member			Object Type Re	-	Not Applicable	Not Applicable
WO Planned Unscheduled		☐	Number	-	Week	Model Calendar
Unscheduled - Basis Member			Object Type Re	-	Not Applicable	Not Applicable
WO planned scheduled		☐	Number	-	Week	Model Calendar
Scheduled - Basis Member			Object Type Re	-	Not Applicable	Not Applicable
Past Due		☐	Number	-	Week	Model Calendar
Past Due WO - Basis Member			Object Type Re	-	Not Applicable	Not Applicable
Capacity		☐	Number	Service Business Unit, s - Simulation, s - Territory L4 - Active	Week	Model Calendar
Overtime Capacity		☐	Number	Service Business Unit, s - Simulation, s - Territory L4 - Active	Week	Model Calendar

10. Use time range as much as possible

Modules that hold history can be separate from the modules that do not. It would be effective to use time range in such cases. Cut down the module size by using only past years for history module and current year for the planning module and future years for AOP and Forecasting modules. Not all the modules need all the time range and hence this feature is always the easiest to enable model optimization.



	Start Period	Number of Periods	Quarter Totals	Half-Year Totals	Year To Date	Year To Go	Total of All Periods
Present years	FY18: 1 Jan 2018	1	☑	☐	☐	☐	☐
Past 2 years	FY16: 28 Dec 2015	2	☐	☐	☐	☐	☐
Total Years	FY16: 28 Dec 2015	3	☐	☐	☐	☐	☐

Conclusion

Model space optimization is not just a science but also an art that requires careful consideration and execution. Optimization should not be an exercise post implementation but should go hand in hand with implementation. Proper check on time settings, evaluating usage of custom lists for different versions, POC approach, hierarchy pruning, managing dimensionality, controlled use of subsidiary view, quality check on line items, maximum usage of time range are the best practices that deserve incorporation in the model to bring it to its optimum size. Business consultancy and customer convincing skills of architect also play an important role in this process. In short, keep it simple!

About the author

Mahesh Damani, Manager - Anaplan Practice, Wipro

Mahesh has close to 14 years of experience in EPM suite of products. He is PMP® and TOGAF® certified and has sound knowledge on requirement gathering, architecture, system design and green field implementations.

Wipro Limited
Doddakannelli, Sarjapur Road,
Bangalore-560 035, India

Tel: +91 (80) 2844 0011
Fax: +91 (80) 2844 0256
wipro.com

Wipro Limited (NYSE: WIT, BSE: 507685, NSE: WIPRO) is a leading global information technology, consulting and business process services company. We harness the power of cognitive computing, hyper-automation, robotics, cloud, analytics and emerging technologies to help our clients adapt to the digital world and make them successful. A company recognized globally for its comprehensive portfolio of services, strong commitment to sustainability and good corporate citizenship, we have over 175,000 dedicated employees serving clients across six continents. Together, we discover ideas and connect the dots to build a better and a bold new future.

For more information, please write to us at **info@wipro.com**

IND/BRD/OCT 2019-SEP 2020