Multi-Dimensional Modeling in SDMX

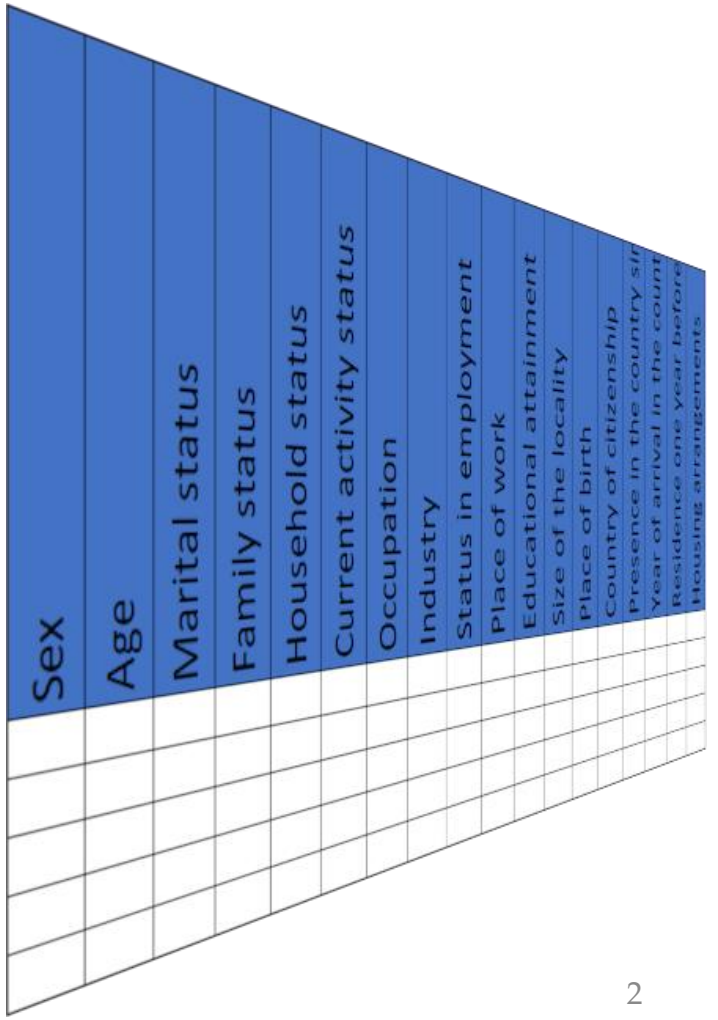
9th SDMX Global Conference

Abdulla Gozalov (UNSD), Matt Nelson (Regnology)

1 Nov 2023

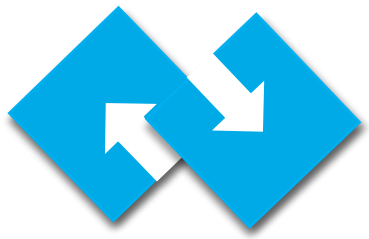
Modeling Highly Multi-Dimensional Datasets

- Working with horizontally complex data structures – i.e. those with many dimensions – is a long-standing problem in official statistics.
- Large number of dimensions in a dataset implies high likelihood that additional dimensions will be required in the future.
- Once finalized, updates to a data structure tend to be expensive since they must propagate to upper tiers of the system.
- In addition, highly multi-dimensional structures usually result in a sparse hypercube
  - Many or most dimensions are usually inapplicable to any given observation



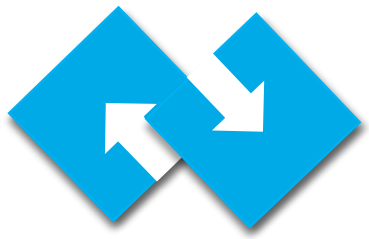
Highly Multi-Dimensional Datasets in SDMX

- Common challenge, particularly in demographic and social statistics
- Extremely high cost of updating reporting data structures since the updates propagate to all reporters causing them to update their data mappings
- Different approaches taken in various reporting DSDs
- "Pure" approach: use pure dimensions, clean code lists, and define as many DSDs as required for data exchange
- "Simple" approach: trade horizontal complexity for vertical complexity by combining multiple breakdowns in the same concept/code list
  - Extending a code list is far less costly and disruptive than adding a dimension



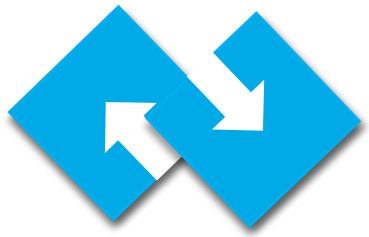
Coping With High Dimensionality: Pure Approach

- Create a separate Data Structure Definition for each distinct hypercube
- Use pure concepts, clean code lists, and only applicable dimensions in each DSD
  - Dense hypercubes
- Characteristic of the European Census Hub
- 60 DSDs in the European Census Hub
  - 60 mapping sets for each reporter to maintain
  - User must navigate the dozens of datasets
- Feasible because the hypercubes are defined in legislation and only change with census rounds, i.e. every 10 years
  - No need to add dimensions



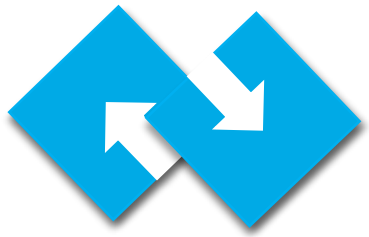
Coping With High Dimensionality: Simple approach

- Use mixed or "wildcard" dimensions
- Use a single concept/code list for multiple underlying breakdowns
- Adding a dimension means extending (or reusing) a code list
  - Converts horizontal complexity into vertical complexity
  - Extending a code list much less disruptive than adding dimensions
- Characteristic of the EcoFin, SDG, and other DSDs



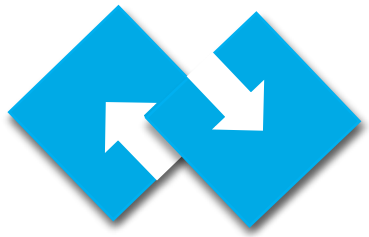
Mixed Dimensions: Examples

- Composite indicators/series code lists with embedded breakdowns
  - **BK_USD** Balance of Payments, Capital Account, Net, US Dollars
  - **BK_CD_USD** Balance of Payments, Capital Account, Credit, US Dollars
  - **BK_DB_USD** Balance of Payments, Capital Account, Debit, US Dollars
- Composite Breakdown: combine many breakdowns in a single code list
  - **FCC_H** Frequency of Chlorophyll-a concentration: High
  - **FCC_M** Frequency of Chlorophyll-a concentration: Moderate
  - **FIS_POSTFIS_CON_INC** Fiscal intervention stage: Postfiscal consumable income
- Custom Breakdown: use generic codes whose semantics are defined at transmission time rather than structure design
  - **C01** Custom code 01
  - **C02** Custom code 02
  - **C03** Custom code 03



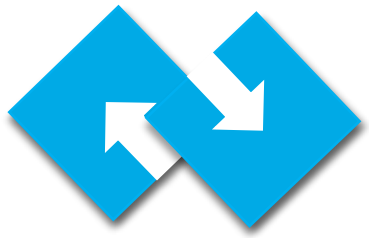
Mixed Dimensions: Drawbacks

- Vertical complexity comes with highly undesirable side effects
  - Proliferation of codes in composite indicator code lists
    - Cartesian product of indicators and any applicable disaggregation implemented in the same code list
  - Potential collision in composite breakdown code lists
    - Cannot use more than one breakdown at a time from the same code list
  - Inability to use standardized codes or persistent identifiers
    - Composite indicators or breakdowns map to several underlying codes
  - Unpredictable assignment of breakdowns
    - New breakdown can be implemented in one of several mixed code lists
  - Difficulty visualizing and working with mixed dimensions
    - Complex labels, cannot transpose embedded breakdowns, must use complex workarounds for wildcard codes, easy to make mapping errors
  - Reporting structures poorly suited for dissemination
    - Separate structures are often created for dissemination purposes, improving visualization at the expense of interoperability



Information Completeness

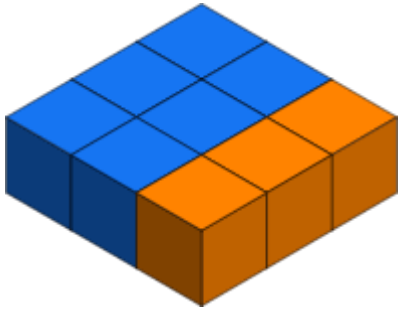
- The high cost of updating data model puts pressure on designers to get the data model right at the first public release
- Reporting structure designers are forced into a waterfall-style design approach
- Cannot start small and grow as required
  - High cost of updating the data model precludes agile development
- Since completeness of information on global data exchange needs is rarely attainable in real world, resulting reporting structures are still imperfect and often difficult to use



Proposed Alternative: Invariant Dataflows

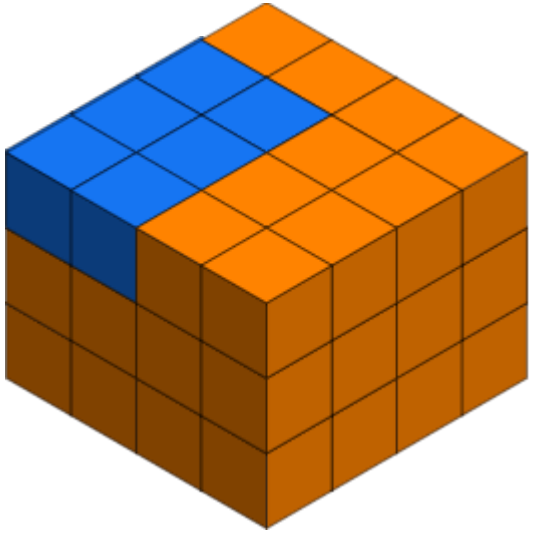
- DSDs are created with pure concepts and clean code lists, regardless of the number of dimensions.
- Dataflows based on the DSD are defined with reduced dimensionality. Only those dimensions relevant for the dataflow are specified, and the rest are unmapped.
- Each reported dataset references a reporting dataflow and only utilizes dimensions defined for the dataflow. Unused dimensions are not present in the dataset.
- Updating the DSD by adding dimensions does not affect existing dataflows, which can continue to be used as is for reporting or dissemination.
- In the tools, mappings are maintained between source data and concepts/codes of the DSD. Updating the DSD by adding dimensions or extending code lists does not affect existing mappings insofar as the new dimensions or codes are not used.

Invariant Dataflow



- A dataflow is created as a slice of hypercube defined by the DSD
- As the parent DSD grows dimensions and codes, the dataflow remains valid and available for reporting or dissemination

Invariant Dataflow



- A dataflow is created as a slice of hypercube defined by the DSD
- As the parent DSD grows dimensions and codes, the dataflow remains valid and available for reporting or dissemination

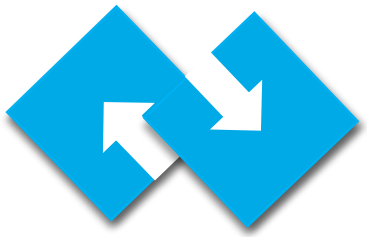
Benefits

- The need for composite or merged code lists reduced or eliminated
- Much faster, agile development of reporting data structures
- Improved interoperability
- Improved visualization
- Simplified structure maintenance
- Easy to maintain, straightforward data mappings
  - Simplified reporting
  - Simplified consumption
- Data structures equally well suited for reporting or dissemination

Implications to the standard

- Dataflows can be defined with reduced dimensionality
  - Using a mechanism such as content constraints, annotations, or another
- Dataset structure is defined by the dataflow
  - Only those dimensions applicable to the dataflow are used in the dataset → partial key
  - Dataset remains valid even as its parent DSD expands dimensions and code lists

- Overall, the effort required to implement the updates appears to be reasonable



THANK YOU!