TECHNICAL CAPABILITIES AND IMPLEMENTATION STRATEGY GUIDE:
RECOMMENDATIONS FOR CUSTOMER CONTEXT DATA

Customer context is much easier to describe than it is to achieve—as demonstrated by the fact that it's not commonplace even though the idea has been around for decades.

We've implemented customer context and related projects in a wide variety of conditions, often coming in on the heels of a project (or projects) that had failed to achieve it before.

From this experience, we're sharing our recommendations for the technical capabilities and strategic approach to data integration that we've found most necessary for a successful customer context project.

RECOMMENDATIONS

TECHNICAL CAPABILITIES

Model

We recommend that you center your customer context technology around a domain-centric model that you use to reflect real-world entities. By using a model as a conceptual pattern to integrate technical functionality, it becomes easier to build a solution that incorporates new data as it arrives.

With the model as a kingpin, you can leverage data clustering, synthesis, and relationship discovery to create customer contexts that are optimal representations of your customers and the relationships between them with very little, and potentially no, non-automated intervention necessary.

Historical Data Retention

Being able to re-evaluate all existing data about an entity when new data arrives is essential to keeping the customer context up-to-date over time and to maximizing the quality of your data.

As some data can be expected to change over time, retaining all data (even data that was not selected for the golden customer context) allows your solution some ability to automatically detect when one of these changes has occurred and update the golden context automatically.

Update Behavior

We strongly recommend engineering your customer context solution with persistence and update requirements in mind. Over time, most customer context will need to be updated. However, as the sophistication and history grows, so does the time it will take to reconstruct the customer context information.

Therefore, it is important to be able to store, retrieve, and update current customer context information efficiently. This can be provided by an architecture where canonical entity information from the customer context graph is only active and involved in processing when it is specifically needed.

Such architecture, if done carefully, can also ensure that your solution's performance for updates is based on the volume of incoming data only, not the volume of existing data; the processing needs of the solution then do not have to grow indefinitely with your accumulated data.

IMPLEMENTATION STRATEGY

When implementing your customer context data integration project, we recommend a data reduction funnel that accepts all possible data and gradually distills it through a variety of processes to produce the final context.

As previously mentioned, new data should pass through the data reduction funnel in real time, while a database preserves earlier decisions and all pre-existing data for re-evaluation.

STAGE 1

Data Ingest

Establish the technical connection between the source systems and your project. The primary activities at this stage are record and field parsing and mapping of the resulting fields to the domain model.

STAGE 2

Filtering and Cleansing

Work with the raw data to make it appropriate for consumption and comparison to existing data. This can involve quality-checking, filtering, cleansing, transformations, and more.

STAGE 3

Apply Model

Apply the relevant parts of the business model to the data resulting from stage 2, using the mappings established in stage 1. Raw data is associated with the expected objects and relationships in the model, and data fragments are created.

STAGE 4

Clustering

Cluster the fragments created during the previous step into groups with any new or existing fragment(s) that represent the same customer. Machine learning can be useful at this stage, especially distance-based clustering algorithms such as k-means. This stage is where match rules representing the different circumstances in which data is considered “matching” and the degree of acceptable variation are used.

STAGE 5

Synthesis

Synthesize each fragment cluster into the best possible representation of the customer in the model. This stage is where the factors such as age, frequency, and trustworthiness of the source system that determine what “best” means in any given circumstances are considered.

For each attribute, implement tuple processing that can consider information from other attributes’ tuples, other attributes’ final state, and/or enrichment data sources.

STAGE 6

Relationship Discovery

With individual entities synthesized, relationships between entities are identified and established. If customer context data about another entity is present, the clustering and synthesis processes can be used to identify and establish relationships with other customer contexts.

BENEFITS

There are a number of benefits to using our recommended approach.

Maximized Preservation of Meaning

This approach naturally maintains the organic connections and relationships between data. It also preserves the lineage and history of your data about a customer.

Better Information Transfer

By structuring the data to mimic its real-world relationships, you can transfer data about a customer in response to a query or as output to another system very effectively, when compared to other integration approaches that cache data based on the load order or other meaning-agnostic factors.

Faster Updates

Data propagation becomes very precise when only entities which data is relevant to are active, and only the parts of the entities that the data changes are modified. Update speed is therefore dependent only on the amount of incoming data, not the total amount of data stored, and small changes to large entities don't bog down the system. We have seen this strategy allow real-time customer context updates even for petabytes of stored data.

The actual speed at which data is updated, retrievable, and/or pushed to other systems can vary considerably based on the resulting or method of retrieval you implement, as well as other solution design decisions that you make. In our implementations, we expect transactional updates to be complete within 500 milliseconds to 2 seconds in very large solutions.

More Flexible System Integration Options

The domain-centric approach should not assume anything about the source of the data. During development, this allows you to use the trailblazer approach to grow the project sustainably. Our strategy also permits sources to be integrated and removed even after the solution is in production, ensuring that your customer context solution can continue to operate even if your IT landscape changes.

CONNECT WITH MIO

Find out how we can help with your customer context project, or just let us know what you thought about this strategy guide.

Joshua Herritz
joshua.herritz@miosoft.com
608-210-1193