Summary Price Match engine

The requirements below align with the Visio diagram titled “Dynamic Pricing

Flow” . The proposed suite of applications is comprised of 3 primary components:

Price Scraping

Price Recommendation

Price Execution

Price Scraping

The Price Scraping application is the vehicle that collects competitive item and pricing

data from outside websites via a user-defined set of rules. The application also includes

rules for matching collected data to internal items. In cases where an exact

match cannot be confirmed the application also provides the ability for user review and

correction. Once items are matched those relationships can be re-used in repeat scans

of the competitor site.

Price Recommendation

The Price Recommendation application uses the data collected via scraping in

conjunction with internal data to develop price change recommendations. The

recommendations are based on a flexible set of user-defined business rules that control

the response to the competitive data. These can include margin limits, price within x%

of competitor, max and min price change limits and a variety of others. The rules can be

defined at any level of the product hierarchy and by competitor and can be re-used

once established. Price Optimization is component of this and can be used, if selected,

in an immediate response.

Price Execution

The Price Execution application will be developed within the New Price

Management system. The application will accept the recommendation data and, based

on rules, either allow the price recommendation to flow directly to or to a

screen for user review / approval. This application will also present exceptions where

the Price Recommendation application failed to create a recommendation due to

current pricing or not all rules being satisfied. The user will also be able to enter a price

change through this application that could then be fed directly to company. The

“immediate” price changes will also be fed into the existing Price Management

system to be processed in the nightly batch to maintain synchronization of price across

the systems.

High-level Requirements

1. Price Scraping

1.1. System has a maintenance ability to control various aspects of web scraping

1.1.1. Product type using current hierarchy

1.1.2. Price range

1.1.3. Competitor sites to search combined with hierarchy information

1.1.4. Competitor sites per geography

1.1.5. Frequency of search refresh

1.2. The system can support matching web scraped product to corresponding

product via a set of rules define at any level of the hierarchy

1.2.1. Rules can leverage price, attributes, description

1.2.2. Rules can be defined per competitor

1.2.3. Success criteria can be defined at any level that determine if a match was

completely successful or requires user-intervention

1.2.3.1. The system provides a set of screens to review matches and

approve / correct or delete

1.2.4. Item match data per competitor is stored and can be used again during a

refresh of competitor prices

1.2.5. Items can be matched to more than one competitor

1.2.6. The system will support the upload of pre-matched item data (e.g.

vendor supplied)

1.2.7. The system will be designed to enable collection of multiple price types

(e.g. PROMO)

2. Price Recommendation

2.1. Pricing recommendation is performed in near real-time using a defined set of

rules

2.1.1. Rules are maintained via a set of rule maintenance screens

2.1.2. Rules can be configured to use both regular and promotion price as factors for

recommendations

2.1.3. Rules can be set with a priority within a set of rules to control which rule

wins out in a conflict

2.1.4. A set of rules can be setup for a range of product (rule model) and will

then be used any time a product that is contained within that range comes

into the system

2.1.5. Geography will be a useable attribute should go to regional

pricing

2.1.6. Items can be selected based on eCommerce status (online only or

online/in-store)

2.1.9. Price Optimization is a selectable option with accompanying rules

2.2. The following are examples of acceptable rules

2.2.1. Product Inclusions / Exclusions

2.2.1.1. Select a product hierarchy level (e.g. division, category) to include

or exclude

2.2.1.2. Provide a specific item list to include or exclude

2.2.1.3. Specify item exclusions based on status (ex. clearance, no sales,

seasonal, DSD, pre-priced, new item)

2.2.1.4. Specify by season code, brand or DUNS

2.2.2. Comp Price Rule

2.2.2.1. Set pricing boundaries against specific competitor’s price (ex.

+5%, -3% to +5%)

2.2.2.2. The tool should be capable of evaluating multiple pricing

guidelines against multiple competitor prices, and resolve conflicts.

2.2.2.2.1. Example: Price from 0 to +5% of the Target price. If our

current price is 10% below the Target price, set new price as

Target price. If current price is 10% higher than Target price, set

new price as +5% of Target price.

2.2.3. Item Relationships

2.2.3.1. Brand Relationships

2.2.3.1.1. Example: How do we price private brands vs. national brands

on similar items?

2.2.3.2. Size Relationship (based on “unit of measure” metric)

2.2.3.2.1. Example: How do we price the large size vs. the small size of

the same item?

2.2.3.3. Other relationships:

2.2.3.3.1. Based on product attributes (ex. set pricing differentials based

on item color)

2.2.4. Price Point Rule

2.2.4.1. Round recommended price to a re-defined set of price points.

Should be flexible to round up or down.

2.2.4.1.1. Example: Every price should be rounded up to a $.99 ending.

2.2.4.1.2. Example: Can have a rounded price based on a price range. If

the recommended price is between $1.00 and $1.06, round to

$1.09.

2.2.5. Maximum Price Move Rule

2.2.5.1. A cap set to limit price moves. Must be able to set different value

for increase and decrease.

2.2.5.1.1. Example: A price increase must be smaller than 10%; a price

decrease must be smaller than 15%

2.2.6. Minimum Price Move Rule

2.2.6.1. A cap set to avoid small price moves. Must be able to set

different value for increase and decrease.

2.2.6.1.1. Example: A price increase must be larger than 1%; a price

decrease must be larger than 2%

2.2.7. Minimum Margin Rule

2.2.7.1. A rule set to restrict minimum margin %.

2.2.7.1.1. Example: Maintain minimum margin of 15% for each item.

2.2.8. Maximum Margin Rule

2.2.8.1. A rule set to restrict maximum margin %.

2.2.8.1.1. Example: Maintain maximum margin of 60% for each item.

2.2.9. Exception Rules

2.2.9.1. Flexibility must exist to turn exception rules on and off.

2.2.9.1.1. Exclude Price Adjustments:

2.2.9.1.1.1. Exclude price adjustment if a price change occurred in

the past XX days/hour/minutes

2.2.9.1.1.2. Exclude price adjustment if a future price change is

pending to occur in the next XX days/hour/minutes

2.3. Data sourced into system will include

2.3.1. SPRS sales data

2.3.2. CORE cost data (based on media code)

2.3.3. Omniture clickstream data to support forecasting

2.3.4. Current prices from Price Management

2.3.5. Prices from eCommerce (includes today’s immediate changes)

2.4. Issues

2.4.1. How will collisions be resolved?

2.4.1.1. Pending normal changes could override immediate changes

2.4.1.2. Promotional prices could use incorrect regular price if an

immediate regular price change is in process

2.4.2. May want to time-box when changes can be sent to eCommerce

3. Price Execution

3.1. Price recommendations will be sent from recommendation engine to New Price

Management application

3.1.1. Screens will allow review of pending changes

3.1.1.1. The system will require user review of prices that exceed defined

tolerances

3.1.1.1.1. If % price adjustment is more than XX%

3.1.1.1.2. Price recommendation rules failed to produce a result

3.1.1.1.3. Others

3.1.2. Changes within defined tolerances will be allowed to go immediately to

site without review. This will be customizable per model.

3.1.3. All changes must be done at the Price Link level

3.1.4. Review will allow all or some recommendations to be approved

3.1.4.1. Two types of review will be definer by user-defined rules

3.1.4.1.1. Passive – price will go without intervention in X hours

3.1.4.1.2. Active – Price is held until user takes action

3.1.4.1.3. Both types of action can result in a user message (email or

mobile)

3.1.5. Urgent overrides can be entered via these screens without input from

recommendation engine

3.1.6. All price changes related to price scraping will be tagged as such for

reporting purposes

3.1.7. eCommerce and Pricing need to determine how to present a SAVE STORY

when prices are changed

3.2. New Price enhancements

3.3. Reporting

3.3.1. Financial impact report

3.3.2. Orphan item report (items that are not captured in any rule model)

3.3.3. Failed recommendation report

3.3.4. Price change dashboard (includes queue reporting for changes)

3.3.5. Others

3.3.6. All reporting will be online

3.4. Synchronization

3.4.1. Immediate price changes sent to engine will also produce a price

change transaction to be processed in nightly Price Management run

3.4.2. Nightly Price Management run will continue to send daily file to

Sears.com – immediate changes will catch-up and be in sync following

nightly batch