Multimedia Integration Pattern Representations

Daniel Ritter
SAP SE
daniel.ritter@sap.com

Stefanie Rinderle-Ma
University of Vienna
stefanie.rinderle-ma@univie.ac.at

***Abstract*—In this document we collect and briefly explain the logical pattern representations for our contribution "Toward Application Integration with Multimedia Data" as well as modeling stereotypes for the user interaction.**

I. Introduction

Tweleve messaging patterns were identified as especially relevant for multimedia application integration. These patterns are discussed with respect to their operations on the physical and logical operations on the messages in Sect. II. Then the patterns are grouped according to user interaction stereotypes and briefly discussed with respect to modeling and configuration in Sect. III.

II. Multimedia Pattern Representation

To show the comprehensiveness of the proposed logical representation, we subsequently discuss the relevant patterns: Channel Adapter, Content-based Router, Message Filter, Aggregator, Content Filter, Content Enricher, Feature Detector, Image Resizer, Idempotent Receiver, and Message Validator in the context of multimedia processing. The patterns are grouped by similar operations on the physical and logical representation as shown in Tab. I.

TABLE I
Integration pattern Multimedia Aspects (logical Log, physical Phy, re-calculated recal)

Pattern Name	Multimedia Operation	Arguments	Phy	Log
explicit				
Channel Adapter	format conversion	format indicator	write	create
Splitter	fixed grid, object-based	grid: horizontal, vertical cuts; object	create	recal/write
Router, Filter	select object	object	-	read
Aggregator	fixed grid, object-based	grid: rows, columns, heights, width	create	recal/write
Translator, Content Filter	coloring	color (scheme)	write	recal/write
Content Enricher	add shape, OCR text	object, shape+color, text	write	recal/write
Feature Detector	segmentation, matching	object classifier	read	create
Image Resizer	scale image	size: height, width	write	write
extra				
Idempotent Receiver	detector, similarity	object for comparison	-	read
Message Validator	detector	validation criteria	-	read

A. Physical write, logical create

The Channel Adapter [1] transforms the incoming message into the physical message representation of the integration system (e. g., multimedia format). From the physical representation, it creates the logical object model (by schema) as conceputally shown in Fig. 1.

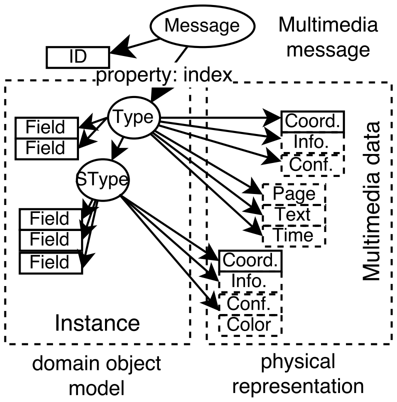

Fig. 1. Conceptual Object Model.

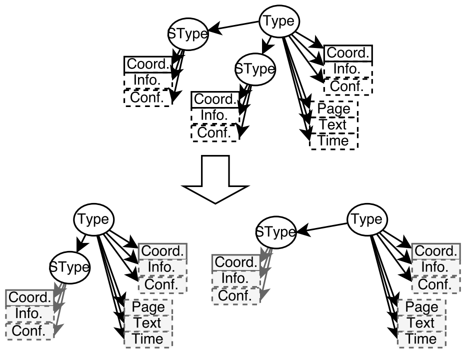

Fig. 2. Splitter.

B. Phyiscal create, logical re-calculate or write

The splitter [1] splits on multimedia message, e. g., with one image, into several multimedia messages either by fixed grid (e. g., equi-distant slices) or by domain obbject (e. g., for each human). While the physical message representation has to be created for each iterable, the logical representation could be re-calculated bassed on the knowledge about the cutss. If this is not possible, the features have to be detected again and the logical model has to be updated. Figure 2 shows the logical representation, before and after the split.

In contrast, the aggregator [1] collects several multimedia messages and combines them to one physical message by

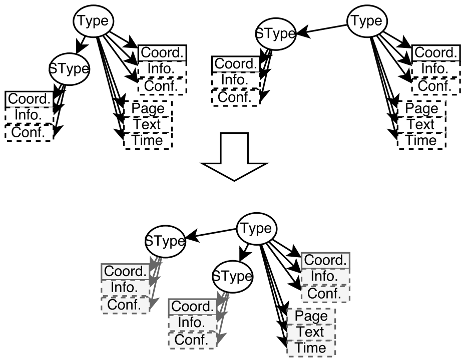

Fig. 3. Aggregator.

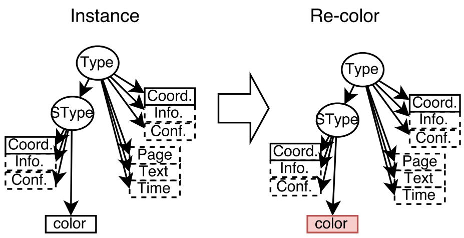

Fig. 4. Message translator, content filter: re-color.

aligning them in a grid (depending on the number of messages and their media sizes). Similar to the splitter, the logical representation shown in Fig. 3 can either be re-calculated or has to be re-detected and written.

C. No phyiscal operation, logical read

The content-based router, message filter, idempotent receiver (all from [1]) and message validator [3] do not change the physical message representation. They read the logical representation as in Fig. 1 using a selector-based access pattern. Thereby they use different conditions, e. g., scontains human, similarity operators for matching duplicate images.

D. Physical write, logical Re-calculate, write

In the identified scenarios and apps, the message translator and the content filter (both [1]) were used to change the color of an image. Therefore the physical message has to be changed and the logical representation can be re-calculated as shown in Fig. 4.

Similarly, the content enricher [1] that is used to add OCR text or geometrical shapes to media content Fig. 5 changes the physical message representation. The additional element can mostly be appended to the logical model (i. e., recalculated). In contrast to the translator, a new node is added, which gets the same base elements of a normal type node.

In all of these cases, the logical representation is changed by overwriting or adding values. These changes are detected and the media content is changed accordingly.

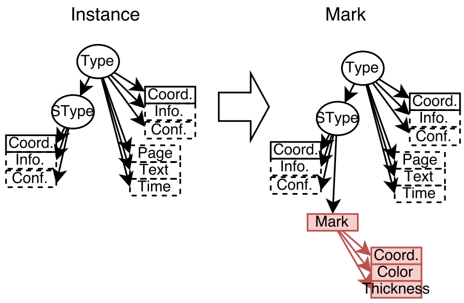

Fig. 5. Content enricher: Add Shape.

E. Physical and logical write

The image resizer [3] changes the physical representation as well as the logical represenation, since a re-calculation of the feature placements is usually not possible (e. g., humans could be scaled so that they are no longer detectable as such). Instead, the image is re-sized and then the features are detected again.

III. Pattern Modeling Stereotypes

These patterns can be grouped as modeling stereotypes from a user interaction view. We differentiate and subsequently discuss the modeling stereotypes *Simple Media Routing*, *Media Routing Decision*, *Split Media*, *Media Merge*, *Simple Operation* and *Operation with external knowledge*. Figure 6 shows the different modeling stereotypes in BPMN according to [2].

A. Simple Media Routing

The *Simple Routing* stereotype denotes the simple modeling for the identified *Capture / Share* case. Thereby messages are received and simply forwarded. The Channel Adapter would be represented by a BPMN Message Flow according to [2] (not shown).

B. Media Routing Decision

The routing decision uses a BPMN Gateway with BPMN Sequence Flow conditions (`cond.` in Fig. 6) on multimedia messages. These conditions are formulated as Selector access pattern that evaluates the logical pattern representation (e. g., object domain model).

With the routing decision stereotype, the Content-based Router and the Message Filter patterns can be realized. Furthermore, the Message Validator can be realized.

C. Split Media

The *Split Media* stereotype, realizing the Splitter pattern, requires an expression (`cond.`, either in form of a selector access or a grid for splitting the media message content. The splitter itself is denoted by a BPMN Task. As discussed in Sect. II, the splitter creates new multimedia messages, that are represented by a BPMN Message Collection.

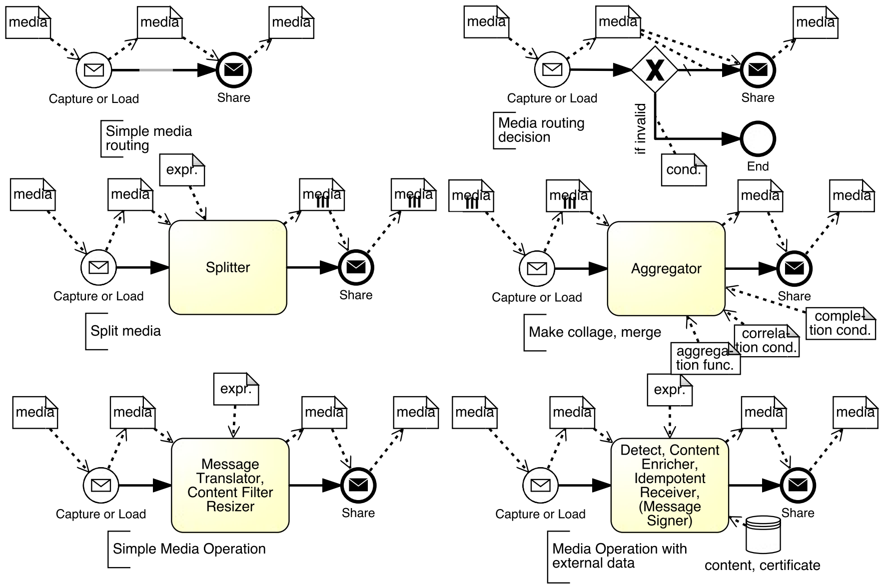

Fig. 6. Modeling Stereotypes for Multimedia Application Integration.

D. Make collage, merge

Similar to the Split Media stereotype, the *Make colalge, merge* is denoted by a BPMN Task. However, it requires a BPMN Message Collection as input and emits a BPMN Message. Therefore, the correlation (e. g., for grouping humans with specific emotions) and completion (e. g., for completing the Message Sequence) conditions are represented by a selector access pattern. The aggregation function is grid-based.

E. Simple Media Operation

As described in Sect. II, the Message Translator, Content Filter, and Image Resizer require an expression that changes the logical representation. Therefore, the *Simple Media Operation* uses a BPMN Task.

F. Media Operation with external data

The *Media Operation with external data* stereotype covers all patterns that require external data. In Fig. 6, the external data is depicted by a BPMN Data Store. Alternatively, a BPMN Service Task could be used to highlight the case that the data could be queried from external sources as well. For instance, the Feature Detector requires the classifiers and domain object models to detect the features and create the logical representation. The Content Enricher is a special case, since it either enriches the message an external data source, or from the message itself. Consequently it could be modeled as Simple Media Operation as well. The Idempotent Receiver pattern requires the external data to compare the current logical representation with those of the previously processed messages.

References

[1] G. Hohpe and B. Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.

[2] D. Ritter. Experiences with business process model and notation for modeling integration patterns. In J. Cabot and J. Rubin, editors, *ECMFA*, pages 254–266. Springer International Publishing, 2014.

[3] D. Ritter, N. May, and S. Rinderle-Ma. Patterns for emerging application integration scenarios: A survey. *Information Systems*, 67:36–57, 2017.