Original publication available at 10.1109/EDOC.2017.23

Toward Application Integration with Multimedia Data

Daniel Ritter
SAP SE
daniel.ritter@sap.com

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

***Abstract*—Traditionally, enterprise application integration (EAI) processes structured data. Recent trends such as social and multimedia computing led to an increase in unstructured multimedia data like images and video streams that have to be processed by EAI. This poses challenges to EAI with respect to variety, velocity, and volume of the processed data. Furthermore, multimedia data has more semantic qualities (e. g., emotions) compared to structured data, making the data processing and user interaction more difficult. In this work, we conduct a literature review of industrial and mobile applications with respect to their usage of EAI in multimedia computing. We derive multimedia operations and map them to the enterprise integration patterns (EIPs). We propose a realization that allows to interact with EAI processes taking the multimedia semantics into account, discuss EAI architecture extensions and study message processing challenges.**

I. Introduction

Through the interest of (business) applications in social media, multimedia, personal and affective computing [29], socio-technical interactions and communications are introduced into applications. Thus, enterprise application integration (EAI) [23], [11] is now required to process unstructured multimedia data, e. g., in agricultural [36], [41], [3], [25] and medical applications [2], and from social sentiment analysis [31], [33], [35]. Following the idea in [20], we argue that the sequence of operations of many multimedia applications actually denote integration processes, thus leading to new EAI characteristics with respect to the representation and variety of the exchanged messages (i. e., multimodal: textual and multimedia), the growing number of communication partners (i. e., message endpoints), as well as the velocity (i. e., message processing styles), and volume (i. e., message sizes) of the data [39].

However, the current EAI foundations in form of the enterprise integration patterns (EIPs) [11] and system architectures [22], [12] do not address the multimedia characteristics. Table I sets the current characteristics of the basic EAI concepts from [23], [11] into context to those of emerging applications [21], [29]. These characteristics lead to the following challenges, which are not met by current system implementations:

(CH1) User interaction and interoperability (interaction with endpoints): (a) the representation of multimodal messages (i. e., relational and multimedia), for instance, in form of message format extensions like attachments (cf. Tab. I), and growing variety of protocols with combined textual and media messages (e. g., seamless integration relational and media processing). (b) The message processing and user interaction changes from relational to multimodal (e. g., conditions, expressions), which (c) requires to deal with semantics in multimedia data, while over-coming the "semantic gap" [21], [37] as in the current MPEG-7 standard. Although this was addressed by several initiatives, they targeted low-level media features that are inadequate for representing the actual semantics for business applications like emotions [32].

TABLE I
Multimedia induced shift of core EAI characteristics

EAI Concept [23], [11]	Foundations [23], [11]	Emerging: Media [21], [29]
Message (definition)	header, body	header, body, attachments
Message Protocol (format)	structured / textual (e. g., XML, JSON)	multimodal: textual, binary / media (e. g., image, video)
Message size	small to medium (B, kB)	medium to large (kB, MB)
Message Endpoint (sender, receiver)	few, static (e. g., on-premise applications)	many, dynamic / volatile (e. g., mobile / IoT devices, cloud applications)
Message channel (transport, style)	asynchronous	synchronous / streaming, asynchronous
Adapter, processor (interaction, processing)	relational	relational, semantic, confidence / probability

(CH2) Architectural challenges: addressing the system design that faces the interaction with a growing number of dynamic endpoints and the co-existence of processing styles: asynchronous and synchronous streaming (cf. Tab. I; solved for textual EIP processing [28]), including additionally required components compared to the current EAI systems.

(CH3) Multimodal processing: combining processing styles (streaming, multimodal), distributed processing units (device, EAI system), process optimizations, and data compression to deal with the increasing message sizes for multimodal content.

For instance, the current implementations of social media sentiment analysis scenarios (e. g., [31]) are either focused on textual information or process multimedia data in a yet ad-hoc way (cf. CH1). As sketched in Fig. 1, they usually collect and filter social feeds from sources like Twitter and Facebook according to configurable keyword lists that are organized as topics. The textual information within the resulting feeds is analyzed with respect to sentiments toward the specified topic. However, many sentiments are expressed by images in form of facial expressions. Therefore, the received feeds would require an multimedia Message Filter, e. g., removing all images not showing a human, an Enricher for marking the feature, a Splitter for splitting images with multiple faces to one-face messages, and an Enricher, which determines the emotional

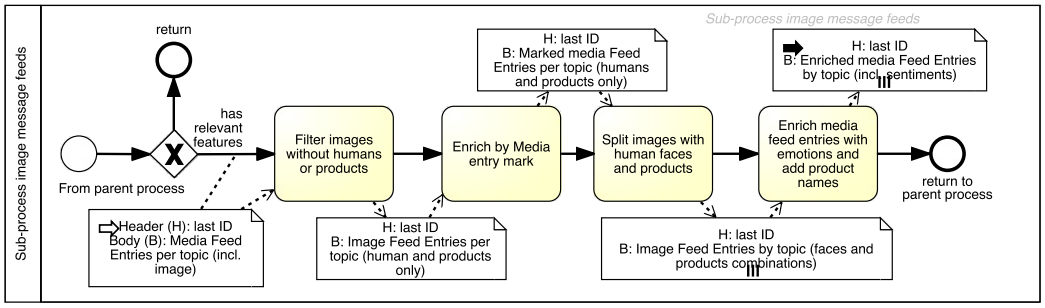

Fig. 1. Multimedia sub-process for social media emotion harvesting (excerpt).

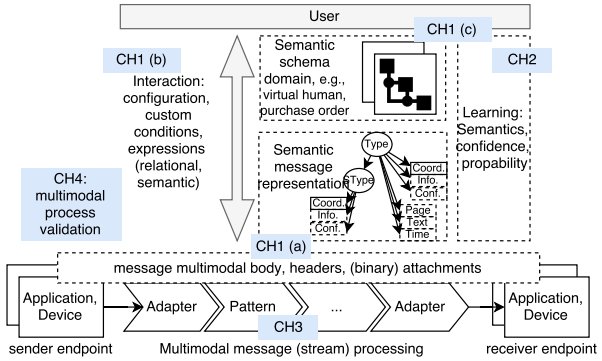

Fig. 2. Challenges of user-centric interaction on semantic message contents.

state of the human and adds the information to the image or textual message, while preserving the image. The interaction with the multimodal messages by formulating user conditions and the required multimedia processing (cf. CH3) are currently done by a large variety of custom functions, thus denote ad-hoc solutions. Therefore, existing EAI systems are extended by – as it seems – arbitrary multimedia processing components in custom projects (cf. CH2) that destabilize these systems and make the validation of the multimodal EAI processes difficult. These challenges are set into context of the current EIP processing in Fig. 2, showing the new problem areas of user interaction (incl. semantic message representation and custom conditions, expressions), new architecture components for learning and detecting the semantics in the multimodal messages, and the multimodal message processing.

In this work, we target answers to the following questions, derived from the introduced challenges.

- User interaction (cf. CH1): Q1 Which industrial and mobile applications require multimedia application integration? Q2 Which integration patterns are relevant? and Q3 How could these patterns be realized and uniformly configured for these scenarios?
- EAI System Architecture Evolution (cf. CH2). Q4 Do the current EAI architectures (e. g., [23], [12]) need extensions to support these scenarios? Q5 Which architectural components are missing?
- Multimodal processing (cf. CH3). Q6 How can the process-oriented multimedia integration processing be realized and improved?

This work does not focus on the areas of content-based media retrieval [21], [4], nor strives to improve existing algorithmic or hardware multimedia processing aspects (e. g., on GPU [34]), but seeks a complementary mapping of the multimedia domain to EAI concepts.

The main **contributions** of this work are a logical representation and physical realization of integration semantics based on an extended EAI system to answer the formulated questions (Q1–6). Therefore, we conduct a scenario analysis in form of a literature and a system review of industrial and mobile applications in the context of the multimedia integration processes in Sect. II (targeting Q1). The analysis results into a mapping of the integration requirements to integration patterns for existing and new patterns, i. e., not in EIP [11] (for Q2). In Sect. III, these patterns are set into context to the integration operations required by the scenarios, resulting to a logical representation toward a uniform user interaction (for Q3). Thereby, we discuss their realization for multimedia integration scenarios (for Q6) and discuss required EAI system extensions in Sect. IV (for Q4+5). The proposed approach is evaluated in Sect. V for its comprehensiveness and message processing throughput for the motivating social media example in a case study (for Q6). We discuss related work in Sect. VI and conclude naming further open research challenges in Sect. VII.

II. Literature and Application Analysis

In this section we conduct a literature and application (app) review targeting Q1 and Q2. The first goal is to compile a list of industries. Based on their scenarios, multimedia operations are discussed that are related to the EIPs from 2004 [11].

A. Methodology

Literature Review. The primary selection of industrial multimedia scenarios from the litereature was conducted using google scholar (scholar.google.com) on 2017-04-03 without patents and citations, for the keyword “image processing industry” and `allintitle`. The search results to 54 articles, of which we selected 29 articles with selection criteria “image processing” (e. g., in abstract, theme) and added the 10 papers as expert knowledge (i. e., examples from the introduction), resulting in 39 articles. We grouped the articles chronologically by the decades they were published and by industry,

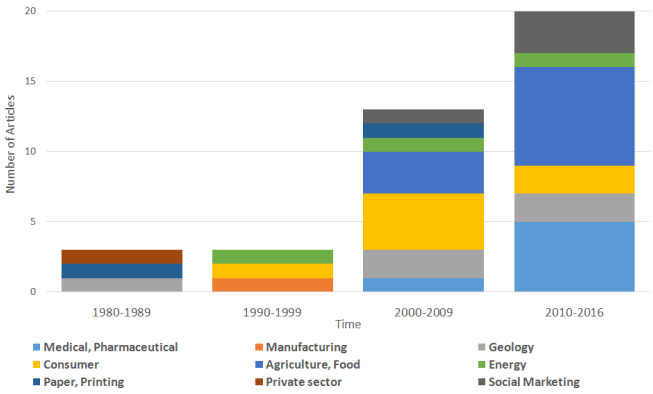

Fig. 3. Literature analysis: Image processing in industries over time

shown in Fig. 3. While the contributions per industry vary, the amount of work found per decade increases. Due to brevity, we subsequently discuss the top three industries from the current decade: agriculture / food, medical / pharmaceutical and social media management. The complete list of 39 selected papers can be found here http://bit.ly/2qkg8RJ.

Multimedia Apps. Similar to [37], we analyze current multimedia apps. The leading app store in terms of the number of applications in 06/2016 is Google Play with 2.2 million applications[1]. Hence, for the application review, we searched in Android Apps with tags "photo", "collage" (e. g., similar to the Aggregator pattern [11]), and "video" with "media" as context by applying the rating "4 stars+" and "for free" filters (e. g., *tags: collage, media*). We considered the first 100 entries and selected those apps with more than one million downloads. As in the literature analysis, the keywords are taken from the problem domain. This resulted in two selections for `tags:+photo,+media`, i. e., Retrica and Instagram, one for `tags:+collage,+media`, i. e., Photo Grid, and one for `tags:+video,+media`, i. e., InShot Video-Editor (only 707k downloads) without duplicates. Conducting a complementing search for a similar "photography" category search adds four more apps, i. e., Google Photo, Snapchat (both image processing), FotoRus (collage) and Textgram (Image+Text).

B. Multimedia Processing Analysis – Results

We consider the media content – found in the literature and apps (e. g., image, text, video) – to be transferred and processed within an integration solution as message body (or attachment; not in EIP [11]) and their metadata as message header. The discussed EIPs denote message processors that base their routing decisions or transformations on the image content (not the metadata).

Following the methodology defined in the previous section, the analysis of the selected literature identified 15 (i. e., nine explicitly and six implicitly named) out of the 48 message processing EIPs as relevant for multimedia processing. Thereby, existing patterns were selected, to which multimedia operations could be semantically assigned. New patterns constitute recurring solutions in form of operations for a specific multimedia data problem (e. g., resize images). Table II shows those of the explicitly named patterns, for which multiple industry or app specific cases were found. The Idempotent Receiver [11] was only named once, thus not shown. The implicitly named patterns (i. e., Datatype Channel, Document Message, Scatter-Gather, Claim Check, Canonical Data Model, Format Indicator) denote basic integration capabilities from [11] that are relevant for all of the multimedia integration scenarios. During the analysis we identified nine new patterns that could not be mapped to an existing EIP, of which two had multiple citations (i. e., Feature Detector, Image Resizer), and thus are in Tab. II. All patterns with only one citation were also identified in [29] (i. e., Message Validator, Message Decay, Message Privacy, Signer, Verifier, and the implicitly named Format Converter) are not shown, however, the validator is discussed further in subsequent sections due to its relevance for multimedia processing.

Literature Review. The results of the literature analysis are summarized in Tab. II. Note that references are added for articles, for which integration patterns could be found. In all of the domains (i. e., agriculture / food, medicine, and social media) the captured images are optionally pre-processed (also mentioned in [20] as "Capture, Share"). The pre-processing usually includes format conversions (e. g., media formats; not shown), resizing (e. g., [9]), message translation (e. g., noise cancellation, consistent background [18], [9], [2]), and content filters (e. g., hair removal for skin cancer [9]). An Image Resizer is also used to compress the data for agricultural monitoring (e. g., fruit monitoring [15], [3], [24]). Alternatively, a Splitter pattern is used to reduce the size of the individual features processed (e. g., [31]). The Feature Detector for semantic objects usually summarizes the low level image processing steps of segmentation, feature extraction and object recognition and can be found in all of the domains. The detected features are then either validated using a Validator (e. g., cancer classification [18], [2]; not shown) [29] or filtered using a Content-based Router or a Message Filter, if they do not have the expected feature (i. e., found in all domains). A Content Enricher is used to add contextual information to images (e. g., weather conditions [36], [41], emotions [31]). Alternatively the image itself is enriched (e. g., by marking faces [31] or suspicious skin moles [2]). The clustering of images using Sequence and Aggregator patterns was found in social media [33]. In [13], the message deduplication is mentioned as removal of "near-duplicates", which is covered by the Idempotent Receiver pattern in [11] (not shown).

Multimedia Apps. These results are backed by the review of the eight mobile multimedia apps, shown in Tab. II. The channel adapters – supported by all apps – denote an important data access or collection facility to capture or load multimedia documents in form of images or videos and share them with the contacts on other social media platforms (e. g., Twitter, Facebook). From the standard routing patterns, only the Ag-

[1]Statista, visited 05/2017: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.

TABLE II
INDUSTRY AND APP ANALYSIS RESULTS (REQUIRED √, NOT REQUIRED (-), PARTLY REQUIRED (√))

Applications	Adapter	Splitter	Router, Filter	Aggregator	Translator	Enricher	Sequence	Detector	Resizer
Multimedia in Industries (from Literature Analysis)									
Farming [3], [15], [36], [41], [26], [6]	√	(√)	√	-	-	√	-	√	√
Medical [18], [9], [2], [24], [17]	√	(√)	√	-	√	√	-	√	√
Social [13], [31], [35]	√	√	√	√	-	√	√	√	-
Multimedia in Mobile Apps (from App Analysis)									
photo+media: Instagram, Snapchat, Google Photo	√	-	-	√	√	√	√	√	-
collage+media: Retrica, FotoRus, Photo Grid	√	-	-	√	√	√	-	-	√
video+media: InShot	√	-	-	√	√	√	-	-	√
text+image: Textgram	(√)	-	-	-	√	√	-	-	-

Abbreviations: Channel Adapter (Adapter), Message Filter (Filter), Content-based Router (Router), Message Translator (Translator) inlcuding Content Filter, Content Enricher (Enricher), Feature Detector (Detector), Image Resizer (Resizer).

gregator was found – especially in the image fusion "collage" apps (e. g., Retrica, FotoRus). Special types of aggregators are the "photo to movie" function in Google Photo and "synchronize music and video" in InShot. Most of the apps make use of special image or video filters, which map to the Message Translator pattern. These filters allow to change all aspects within an image, comparable to the well-known textual Message Translator [11]. The enrichment of images or videos with additional information like layouts, backgrounds, or texts can be seen as content enricher pattern. Notably, Instagram allows to group images as story that vanishes after some time (i. e., Message Decay [29]; not shown). While this denotes a Message Sequence [11] (i. e., single messages belonging together), the aspect of a timed decay of a message or sequence is not in [11]. Similarly, Instagram has self-deleting images, which adds message decay or aging [29].

Further new functionality (i. e., not in EIPs) can be considered Message Privacy [29] (not shown) as "private send" in Instagram, the detection of places or objects and the new processing style of streaming (not shown) in Google Photos, the signing and verification of images as "Retrica-Stamp" in Retrica (i. e., message authenticity [29]; not shown), and the cut, crop or resize capabilities in PhotoGrid, FotoRus and InShot that transform the images beyond the message translator pattern.

C. Summary: Multimodal Pattern Classification

The literature and app reviews identified several industrial domains and mobile apps that require multimedia EAI. We collected the required integration aspects by mapping them to the existing EIP [11] that are affected by multimedia processing as well as identified several new patterns (i. e., Feature Detector, Image Resizer, Validator, Message Decay, Signer and Verifier), which the last four were already found in [29], thus not further discussed here. In contrast, patterns like Wire Tap or Recipient List [11] were not required by the applications, thus do not show any significant relation to media processing. For the subsequent definitions, we classify these patterns according to the dimensions "complexity" and "modality", separating simpler from more complex operations as well as single modal (i. e., textual, multimedia) from multimodal processing (i. e., combined textual and multimedia). Figure 4 depicts these categories that are currently not covered – apart from "Capture, Share" (Adapter Channel) and "Text-to-Text". While many multimedia processing approaches focus on the metadata (e. g., [10]), and thus are "Text-to-Text", "Media-to-Media" denotes an exclusive processing of the multimedia data. Similarly, all complex, but single modal cases are either exclusively textual or multimedia processing (e. g., enrich image by adding geometrical shapes). For some of the complex cases, additional resources are required like a data store for the aggregator or a key store for the Signer pattern [29]. The simple multimodal processing denotes transformations between textual and multimedia (e. g., text to image or image semantics to text). The more complex, multimodal processing includes multimodal operations like the "Media-to-Media,Text" case. We mainly focus on "Media-to-Media", and the routing and transformation patterns from the analysis (e. g., filter, split, aggregate, enrich), required for the identified multimedia integration scenarios.

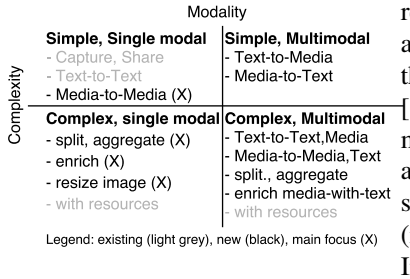

Fig. 4. Multimodal operation classification

III. MULTIMEDIA EAI CONCEPTS

In this section, we map the multimedia operations to the relevant integration patterns from Tab. II (for Q3). Similar to [4], we then define a conceptual, logical representation toward a uniform user interaction (i. e., pattern configuration incl. conditions and expressions) and a physical representation for the evaluation during runtime execution, thus separating the runtime from the user interaction.

A. Integration Patterns in the Context of Multimedia

Table III lists the relevant patterns from Tab. II (by Pattern Name) and sets them into context to their multimedia operations. We focus on the explicitly mentioned patterns in Tab. II (without the Sequence) and include the Idempotent Receiver, Message Validator from the list of the patterns that were mentioned only once. All other non-listed as well as the implicitly required patterns are either covered implicitly (e. g., Scatter-Gather pattern is a combination of the splitter and aggregator patterns) or left out due to brevity. In addition, to the pattern and the corresponding multimedia operation,

TABLE III
INTEGRATION PATTERN MULTIMEDIA ASPECTS (RE-CALCULATED RECAL)

Pattern Name	Multimedia Operation	Arguments	Physical	Logical
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

the (semantic) configuration arguments relevant for the user interaction are added, while assuming that all operations are executed on multimedia messages that are part of the physical representation. For instance, all of the image collage mobile apps in Tab. II require grid-based image fusion for rows and columns or specify height and width parameters. The splitter, required in the social, but also partially in medical and farming industries, either requires simple (fixed) grid-based horizontal or vertical cutting or a more complex object based splitting. Subsequently, we introduce the physical and logical representation, in which contexts the relevant multimedia EAI concepts and patterns are defined.

B. Physical Representation

The basic EAI concepts located in the physical or runtime representation, according to Fig. 2, are the (multimedia) Document Message, Message Channel, Channel Adapter, Message Endpoint (all from [11]), and Format Converter (from [29]). In addition, all identified routing and transformation patterns have a physical representation, with which they interact. These patterns are grouped by their logical and physical data access (cf. Tab. III) as *read/write* and *read-only*.

1) Basic Concepts: For multimedia processing, the physical message representation covers the multimedia format, on which the multimedia operators are applied. Hence it is specific to the underlying multimedia runtime system or library. The current message definition from [11] of textual content (i. e., body) and their metadata (i. e., headers) is therefore extended by binary body entries and attachments. That allows to represent and query structured information in the message body together with unstructured information as attachments at the same time.

As denoted in Tab. III, there are patterns that create, read and change / write to these messages. For instance, the Channel Adapter receives the multimodal messages from a Message Endpoint (e. g., Twitter, Flickr) and transforms (`write`; similar to the Type Converter) the textual and multimedia formats into a physical runtime representation (e. g., JPEG to TIFF for OCR processing) as part of a Canonical Data Model and creates (`create`) the logical representation that is based on the semantic features of the multimedia message content, for the user interaction. However, not all of the current integration adapters are able to handle binary content in the message body and/or attachments (e. g., SMTP separates both, while HTTP only sends one multi-part body).

2) Read/write Patterns: The Splitter splits the multimedia message either by a fixed grid (e. g., cut in half) or based on its domain objects (e. g., human) into smaller ones. Thereby new physical message are created, while the logical representation has to be updated, if it cannot be recalculated (e. g., by exploiting the information on how the image was cut). The aggregator pattern denotes the fusion of several multimedia messages into one. Therefore, several images are combined using a correlation condition based on a multimedia object (e. g., happy customers), and aggregated, when a time-based or numerical completion condition is met (e. g., after one minute or four correlated multimedia messages). The aggregation function denotes a fixed grid operation that combines the multimedia objects along the grid (e. g., 2x2 image frames from a video stream). The logical and physical operations are the same as for the splitter. Similarly, the Translator and Content Filter change the properties of a multimedia object (e. g., coloring). Since this operation is less relevant for business application, it denotes a rather theoretical case, which might only slightly change the logical, however, changes the physical representation. In contrast, the Content Enricher adds geometrical features like shapes to images, e. g., relevant for marking or anonymization, or places OCR text, e. g., for explanation, highlighting. Thereby, the physical and logical representations are changed or recalculated. The Image Resizer scales the physical image and their logical representation, which cannot be recalculated in most cases. The resizer is used to scale down images similar to message compression.

3) Read-only Patterns: The content-based router and message filter patterns base their routing decision on a selected feature or object (e. g., product, OCR text) through reading the logical representation, while the physical multimedia data remains unchanged. Therefore, the features or objects within the multimedia data have to be detected. In the analysis, a separate Feature Detector was required, which reads the physical representation and returns a corresponding logical feature representation. Based on this logical representation, the Idempotent receiver and Message Validator patterns work in a read-only mode.

C. Logical Representation

The logical representation targets the user interaction, and thus defines a Canonical Data Model based on the domain model / message schema of the messages and the access patterns. Due to brevity, we comprehensively list the logical representations for the relevant patterns in a non-mandatory supplementary material http://bit.ly/2qr3hgi, where we also discuss pattern modeling aspects.

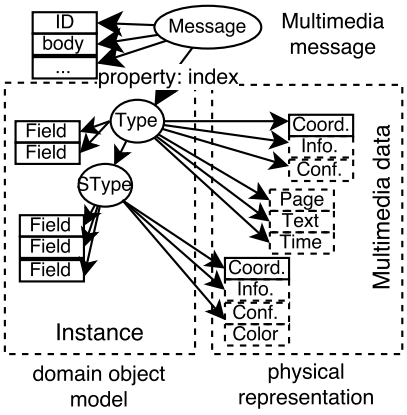

Fig. 5. Conceptual Object Model.

1) Canonical Data Model for User Interaction: While there are standards for the representation of structured domain models (e. g., XSD, WSDL), in which business domain objects are encoded (e. g., business partner, customer, employee), multimedia models require a semantic representation with a confidence measure that denotes the probability of a detected feature. In contrast to [4], who defines a relational multimedia model, we assume a graph structured schema of the domain object (e. g., human expressing emotion) with properties on nodes and edges. Figure 5 depicts the conceptual representation of a property graph starting from the message root node (and its properties, e. g., the message identifier). For the domain object sub-graph (i. e., type `Type` with sub-types `SType`), we add another property to the (semantic) Document Message from Sect. III-B1, which is a transient and removed from the message, before sent to a receiving Message Endpoint. To express the confidence on the detected domain object, all type and sub-type nodes get a `Conf.` field (e. g., type=human with conf.=0.85, stype=emotion, value=happy with conf.=0.95). With this compact definition, lists of arbitrary domain objects can be represented. Through the schema information, these graphs can be formally evaluated. An instance of this model is created during message processing by the Feature Detector pattern (cf Tab. III).

2) From Multimedia Features to Domain Objects / Message Schema.: In our case, the term "Semantic Gap" [21], [32] denotes the difference between low-level image features (usually represented by n-dimensional, numerical vector representing an object, called feature vector) and the actual domain object that has a meaning to the user. According to the scenario analysis, we consider the following image features relevant: color, position, time (interval) in a video stream, during which the domain object was visible or the page number in an OCR document. We assume the creation of the domain object from the underlying features as given by the existing content-based media retrieval mechanisms (e. g., cf. Sect. VI), which is during the message processing in the physical runtime representation. However, for a mapping between the runtime and logical representation, we add the identified image features to our multimedia message index (cf. Fig. 5).

3) Access Patterns: The defined canonical data model is at the center of the user interaction. However, the user should mainly require knowledge about the actual domain model, and thus formulate all integration conditions and expressions accordingly. Subsequently, we identify and discuss common access patterns based on pattern arguments and the logical data access in Tab. III.

Feature Selector. The Content-based Router, Message Filter, Idempotent Receiver and Message Validator patterns as well as the correlation and completion conditions of the aggregator (not shown), the object split condition of the splitter, and the content enricher "mark object" operation are similar in the way they access the data and which multimedia artefacts they require. They require a Feature Detector to detect the domain object (by schema) and create the logical representation. Based on this information the object is selected and the corresponding operation is executed. For instance, the runtime system detects a human and his / her facial expression within an image, using the detector and creates the corresponding message model. Now, the user can configure the splitter to select humans and add conditions for facial expressions, to select them using the selector. Once selected, the splitter cuts the image according to the image coordinates of the selected feature and returns a list of sub-types in the number of humans and the corresponding cut images. The underlying integration runtime system takes the list of sub-types and images and creates new messages for each sub-type / image pair.

Detector Region. The creation of the defined message model through feature detection is computationally expensive, since it involves image processing. Each pattern in an integration process requires such a detect operation, if there is no detector prior to the pattern. Consequently, the detector can be built-in into each pattern or added as separate operation, before a sequence of several *read-only* patterns or patterns, for which the message graph can be re-calculated (e. g., aggregator, splitter; cf. Tab. III). For instance, for fixed grid (with pre-defined cuts) and object splitters, the cut regions are known, and thus the properties of the model type can be computed (e. g., coordinates, color) and does not need to be detected. And the Content Enricher mark operation appends the shape, color, coordinates und a new `mark` node in the graph, thus no detection is required. This way, all subsequent patterns after a detector share the same message property index and do not require further image operations. We call such a pattern sequence Detector Region.

Parameterized Access. Additional information is required for some of the patterns that change the physical representation like the Image Resizer, which requires scale parameters, or the shape and color information for the enricher and the translator. Therefore these patterns modify the feature vector directly (e. g., by changing the color or size). These changes are detected and executed on the physical multimedia object.

IV. Pattern Realization and EAI System Architecture Extensions

In this section, we describe realizations for the described logical and physical representations as well as the resulting

architectural extensions to EAI systems. As EAI system, we chose the open-source Apache Camel [12] due to its broad support of the existing EIPs [29] and its extensibility for new patterns and pattern realizations (e. g., multimedia).

A. Pattern Realization

For the pattern realization, we require the following decisions according to the definitions in Sect. III. Besides Apache Camel as EAI system as part of the physical representation we chose JavaCV (i. e., based on the widely used OpenCV[2] library) as open source multimedia processing system including their type converters. For the feature detection with JavaCV, we use Haar classifiers (e. g., for facial recognition [40]), which has to be trained with positive examples of a features (e. g., faces) as well as negative examples (i. e., arbitrary images without the feature). It is a cascading classifier, consisting of several simpler classifiers that are subsequently applied to an image or region and retrieve the coordinates as well as the object type that can be retrieved. All entering multimedia messages are processed by applying the classifiers.

The logical representation requires a semantic graph, for which we use the W3C Resource Definition Framework (RDF) semantic web standard, similar to metadata representation of images in Photo-RDF (cf. related work). For the schema representations, we chose ontologies similar to [37] that exist, e. g., for humans emotions (cf. vitual human ontology [7]) or real-world business products. For each ontology, a classifier is required to the physical runtime system. The selectors on the semantic RDF graph model are realized by SPARQL queries. The user interacts with the system (cf. Fig. 2) by selecting a schema in form of an ontology and adds the SPARQL query according to the access patterns in Sect. III-C3. If the system has built-in ontology / classifier combinations, only the query is added. Thereby only the domain ontology has to be understood. For parametrized access, our extensions from the physical representation have to be learned by the user.

B. EAI System Architecture Extensions

The system aspects required for the pattern realization, can be summarized to the conceptual architecture building-bocks in Fig. 6. The physical system aspects include multimedia type converters and multimedia libraries. These libraries require feature learning components that learn classifiers for the semantic objects in multimedia data. The libraries evaluated the data according to the classifiers. For the mapping between ontologies and classifiers, the Multimedia Cond., Expr. Evaluation contains the stored domain object models (e. g., ontologies; not shown) as well as the repository for user conditions and expressions (e. g., RDF statements).

For the evaluation of our approach, we extended the existing EIPs in Apache Camel by JavaCV multimedia processing and type converters as well as Apache Jena[3] ontology, RDF representation and SPARQL queries.

[2]OpenCV, visited 05/2017: http://opencv.org/

[3]Apache Jena, visited 05/2017: https://jena.apache.org/

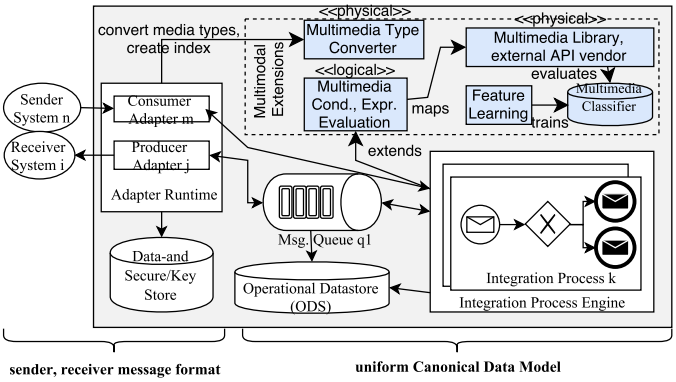

Fig. 6. Conceptual EAI system architecture with multimedia extensions

V. Evaluation

In this section, we evaluate the pattern coverage and comprehensiveness of our multimedia integration pattern realizations from Sect. IV, and apply them to the motivating social media example in a realization and throughput study.

A. Pattern Coverage and Comprehensiveness

The model shall be compact, but comprehensively usable with different image processing systems. Through the separation of the physical runtime and logical representation for user interaction, the comprehensiveness can be checked by its pattern coverage and finding mappings to different image processing systems, while keeping the integration conditions and expressions stable. For this, we selected five multimedia processing systems / APIs from established artificial intelligence vendors: Google Vision API, HPE Haven OnDemand, IBM Watson / Alchemy Services (e. g., also used in [16] for textual analysis and semantic tagging), Microsoft Cognitive Services, and ABBYY, which focuses exclusively on OCR / Text. The complete list of references can be found here http://bit.ly/2ksiqZD.

Pattern Coverage. Figure 7 depicts an overview of the integration patterns that could be implemented by using the vendor systems. We added the open-source multimedia processing libraries OpenCV and Tesseract – used to realize our reference system – for comparison as 0.5 (meaning partially supported due to implementation effort). From our pattern list (cf. Tab. III), the Feature Detector (i. e., for object, emotion, geo, OCR) and the Content Enricher are explicitly covered. While all of the vendors offer object detection and enrichment capabilities in image or OCR texts (e. g., text, face and emotion detection) or geometrical shapes (e. g., Google, IBM), other operations are not supported (e. g., general message translation, resizer, aggregator, security). Therefore extensions in form of custom media processing are usually required, which we realized as integration patterns using OpenCV.

Comprehensiveness. The logical representation in our approach defines the following set of entities, for which a mapping to concepts from the vision APIs has to be found. All multimedia types have a domain object type that is derived

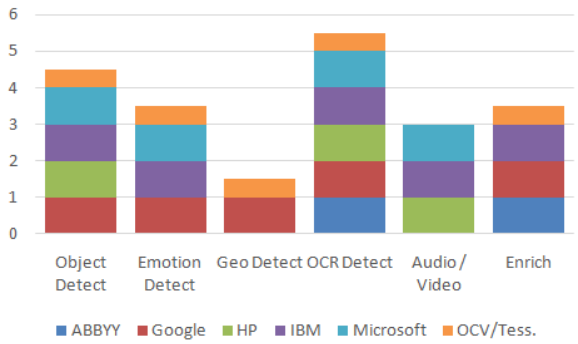

Fig. 7. Vendor libraries as part of an integration system

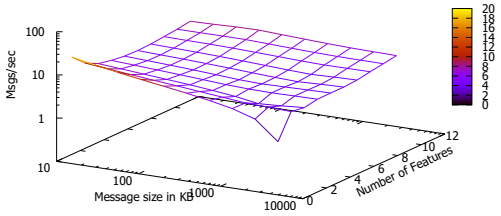

Fig. 8. Multimedia message throughput of the social marketing scenario.

TABLE IV
MODEL COVERAGE COMPARED TO VISION API DEFINITIONS (EXISTS AND MAPPABLE +, NOT SUPPORTED -, PARTLY EXITS OR MAPPABLE +/-)

Vendor / Entity	Image/Any Object type	Coord.	Info	Prob.	OCR Page No.	Text	Video Time
Google	+	+	+/-	+/-	-	+	-
HP	+	+	+/-	+/-	+	+	+
IBM	+	+	+/-	+/-	-	+	+
Microsoft	+	+	+/-	+	-	+	+

from the domain model (ontology), the coordinates of the detected domain object within the medium, object metadata, and the probability for the correctness of the detection. For OCR, the actual text and a page number (for documents) is added, as well as the time (interval) in video streams. Table IV sets our model entities into context to the vision APIs with respect to whether the concept exists and a mapping is possible. Since there was no information for ABBYY, and OpenCV, Tesseract OCR require explicit programming these systems are left out. Although the approaches are diverse in their terminology, provided features and focus areas, the analysis shows a broad coverage for the general model elements. Notably, the object type is represented as list in HP, which we map to different feature vector dimensions. The metadata is mostly provided as name/value (e. g., Google, HP) pairs or tags (e. g., Microsoft). This information is only usable in integration conditions and expressions, if it can be mapped to the model domain. While all vendors add a likelihood (e. g., Google) or score to their models, only Microsoft supports a fine-grain likelihood per feature vector dimension. In terms of compactness of our model, the page number in OCR documents could be left out, since it is only supported by HP. We decided to stick to it for convenience, in case it is available. In summary, our proposed model can be mostly mapped to concepts from heterogeneous vendors and appears compact in its representation.

B. Case Study: Social Media Scenario

In this section, we apply the presented approach on the motivating social marketing scenario presented in Sect. I and show its impact on the message throughput in terms of messages sizes and number of detected features (similar to [30]). Therefore, we extended the open-source integration system Apache Camel [12] by the architecture components in Fig. 6 to realize the multimedia patterns from Sect. III. We discuss the trade-off between message sizes and throughput and compare the normal processing with the Detector Regions from Sect. III-C3. As indicated in Fig. 1, the selector region comprises the content-based router, message filter, translator, splitter and the enricher, for which the logical representation can be re-calculated. For this case study we assume image message workloads from the social media Open Images Project data set based on Google Flickr [19], generated by the EIPBench benchmark tool [30]. For instance, for filtering image messages without a human, we use the SPARQL ASK query `ASK{FILTER NOT EXISTS {?s prefix:hasFace ?o}}`, evaluated using the Apache Jena library, which returns a Boolean that is mapped to the filter runtime component. Similarly, the selector for splitting image messages with multiple humans to single messages with only one is defined as `SELECT ?o WHERE {?s prefix:hasFace ?o}`, returning a list of feature vectors and their coordinates that are then cut and routed separately by our splitter extension.

Figure 8 shows the message throughput of the implemented scenario for an increasing number of features detected in the images and message sizes. Notably, the number of features has less impact on the throughput than the message sizes (corresponding to the image's resolution). Hence, an image resizer or splitter pattern could be used to improve the message throughput, as long as the features can still be detected. For the detector region measurement, a Feature Detector pattern is inserted before the content router. All subsequent pattern are contained in the detector region, and thus do not need to detect the features again. Figure 9 shows the message throughput of the scenario for mixed workload messages size intervals of 1-50 kB, 50-100 kB and 850-900 kB messages with one, eleven, and seven features, respectively. When using the detector region the throughput increases by 2.5% and 10.3% for the smaller message sizes, however, only 0.2% for the bigger message size. While the normal processing is limited by the pattern with the least throughput, the detector region is limited by the throughput of the detector. For larger images the normal and detector region throughput are similar, due to the increasing costs of the feature detection compared to the other pattern processing. Therefore only improved image processing techniques (out of scope), parallel sub-process execution improve the throughput.

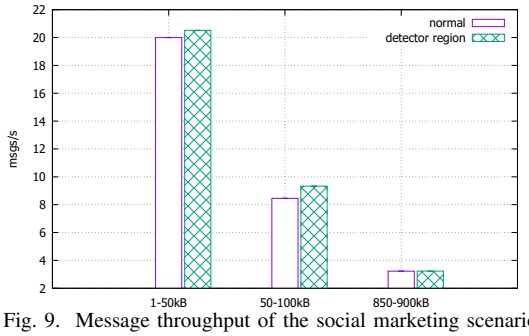

Fig. 9. Message throughput of the social marketing scenario.

VI. Related Work

There is a large body of work produced by research conducted in multimedia processing in venues like ACM Multimedia, ACM Multimedia Systems, IEEE Multimedia, IEEE Transactions on Multimedia. While most of the work targets foundational image processing and feature extraction – complementary to our work – we subsequently set the relevant work into context of our solutions for the challenges in Sect. I.

User Interaction and Interoperability (cf. CH1). The work on queries on multimedia databases and streams targets multimedia data representation and query, similar to our approach. In this context, many user interaction approaches focus on the media's metadata (e. g., name, type, publisher) and not on the actual information within the image (e. g., [10]). Commonly, this metadata is accessible by standards like Photo-RDF[4], which represents the semantic information in images using RDF. While the metadata denotes textual processing, we focus on the multimedia processing. In our realization (cf. Sect. IV) RDF is used to represent multimedia semantics.

Further known related work targets the retrieval of multimedia information from (distributed) databases [4]. The multimedia semantics are represented by semantic attributes based on extended generalized icons with a logical and physical representation on a database. While our approach separates these different representations as well, [4] targets extended normal forms and functional dependencies between different attributes and does not define user interaction with the multimedia semantics on a business application-relevant feature level that could be used for message processing. More recently, Lin et al. developed a similarity query mechanism for images [22] in the area of multimedia queries on multimedia databases. While no query syntax is provided, the operator could be used to formulate decisions based on image similarity.

System Architecture (cf. CH2). Our evaluations in Sect. V identified the need for a change in the common EAI system architectures [23]. While we collected the components required for EAI, we consider the existing work on multimedia processing system architectures complementary to our approach. For instance, there are several systems for parallel media processing. For processing large video streams, is handled using distributed resource management in [39]. Similarly [1] introduces a dataflow process network of actors, connected by FIFO queues, that process multimedia data and fire events based on rules according to a domain-specific metamodel.

The OCAPI system was developed for the semantic integration of programs using a knowledge base approach including a query processor and reasoner over image data for syntactic and semantic integration. The knowledge base is used similar to the ontologies in our approach – giving a semantic context, while the query language works on the image primitives, thus rather technical. No standard query mechanism is provided, however, the R^*-based indexing technique might be considered for optimizing the image message processing. The EADS *WebLab* project denotes a service oriented architecture for developing multimedia processing applications [8]. It neither targets integration processes, nor the EIPs, but defines an exchange format based on a *Media Unit* to solve the problem of semantic interoperability between the information processing components. Similar to our approach, the media types image, OCR text and video are distinguished, coordinates are specified, and a temporal segment is defined for videos. The query approach is based on a proprietary model.

In the related business workflow domain, [27] define the ARIA system with quality of service (QoS) guaranteeing multimedia workflow processing. The defined media filter and fusion multimedia operators in ARIA are similar to our message filter and aggregator patterns. However, the processing is limited to simple 1:1 and fork 1:n workflows.

Multimedia Processing (cf. CH3). The recent survey on event-based media processing and analysis addresses approaches and challenges in the domain of multimedia event processing considering audio, video and social events [37]. Events are human actions or spacial, temporal, relationship state changes of objects, which are mostly represented in event or situational calculi as well as contextual ontologies. Similarly, we use domain-specific ontologies to represent the message schema in our realization (cf. Sect. IV). The app analysis is based on mobile apps like Flickr. The challenges name the discussed "Semantic gap" as well as a "Model Gap", which is the trade-off between an event model's complexity and its detection performance, which we discussed as part of our evaluation (cf. Sect. V).

Overlapping with the challenges of interoperability and system architecture, [5] defines an interoperable interface for distributed image processing using grid computing based on CORBA object exchange. However, the interface and the operations target low-level image processing (e. g., for point and image arithmetic operations). [14] defines a system that segments and indexes TV programs according to their audio, visual, and transcript information. However, the approach uses a "Media-To-Text" preprocessing, while subsequent operations are then executed "Text-to-Text". More recent work on parallel processing of multimedia data mining for computer vision uses map-reduce techniques [38] or cloud-based hadoop systems [42]. The solutions provided target the efficient multimedia program execution on a lower level (e. g., edge detection and segmentation), which could be considered for more efficient message processor implementations.

[4]W3C - Photo-RDF, visited 05/2017: http://www.w3.org/TR/photo-rdf/

VII. Conclusion

In this paper, we address the fundamental topics of multimedia application integration and provide a solution toward a more standard user interaction and configuration of multimedia scenarios. We conducted literature and application studies to identify industrial and mobile scenarios requiring multimedia integration, which resulted to a list of patterns (mostly in [11], [29]) relevant for multimedia processing (cf. Q1+2). For the underlying integration semantics of these patterns we defined multimedia pattern realizations, to which we mapped the operations from the analysis (cf. Q3). We outlined a compact logical, multimedia representation – toward a uniform user interaction that takes the image semantics into account – evaluated the compactness and comprehensiveness by comparison with a selection of vision API vendors. For multimedia processing, the common architecture of EAI has to be extended (cf. Q4). We discussed the fundamental components (cf. Q5) and conducted a case study based on the motivating social marketing scenario (cf. Q6).

Thereby we identified further challenges targeting more efficient message processing (e. g., read / write optimizations like message indexing, process optimizations), interactions with non-standard message transformation operations (e. g., image resizer), new processing types compared to the EIPs (e. g., streaming [29]) and definition of visual integration scenario editors (e. g., query by sketch / visual queries).

Acknowledgments: We thank David Hentschel for his implementation support with the multimedia pattern realizations.

References

[1] X. Amatriain. A domain-specific metamodel for multimedia processing systems. *IEEE Transactions on Multimedia*, 9(6):1284–1298, 2007.

[2] R. Amelard, J. Glaister, et al. Melanoma decision support using lighting-corrected intuitive feature models. In *Computer Vision Techniques for the Diagnosis of Skin Cancer*, pages 193–219. Springer, 2014.

[3] M. Bhange et al. Smart farming: Pomegranate disease detection using image processing. *Procedia Computer Science*, 58:280–288, 2015.

[4] S. K. Chang, V. Deufemia, G. Polese, and M. Vacca. A normalization framework for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1666–1679, 2007.

[5] A. Clematis, D. DAgostino, and A. Galizia. An object interface for interoperability of image processing parallel library in a distributed environment. In *ICIAP*, pages 584–591, 2005.

[6] Q. Dai, D.-W. Sun, J.-H. Cheng, H. Pu, X.-A. Zeng, and Z. Xiong. Recent advances in de-noising methods and their applications in hyperspectral image processing for the food industry. *Comprehensive Reviews in Food Science and Food Safety*, 13(6):1207–1218, 2014.

[7] A. García-Rojas and et al. Emotional face expression profiles supported by virtual human ontology. *Journal of Visualization and Computer Animation*, 17(3-4):259–269, 2006.

[8] P. Giroux et al. Weblab: An integration infrastructure to ease the development of multimedia processing applications. In *ICSSEA*, pages 129–138, 2008.

[9] S. Gopinathan and S. N. A. Rani. The melanoma skin cancer detection and feature extraction through image processing techniques. *IJETTCS*, 5(4):106–112, 2016.

[10] W. I. Grosky. Managing multimedia information in database systems. *Commun. ACM*, 40(12):72–80, Dec. 1997.

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

[12] C. Ibsen and J. Anstey. *Camel in Action*. Manning Publications, 2010.

[13] M. Imran, C. Castillo, et al. Processing social media messages in mass emergency: A survey. *ACM Comput. Surv.*, 47(4):67:1–67:38, 2015.

[14] R. S. Jasinschi, N. Dimitrova, T. McGee, L. Agnihotri, J. Zimmerman, and D. Li. Integrated multimedia processing for topic segmentation and classification. In *ICIP*, volume 3, pages 366–369, 2001.

[15] M. Jhuria, A. Kumar, and R. Borse. Image processing for smart farming: Detection of disease and fruit grading. In *ICIIP*, pages 521–526, 2013.

[16] J. Jovanovic, E. Bagheri, J. Cuzzola, D. Gasevic, Z. Jeremic, and R. Bashash. Automated semantic tagging of textual content. *IT Professional*, 16(6):38–46, 2014.

[17] Y. Juan-ning, W. Ying-zhuo, and Y.-y. ZHANG. Dsp-based image processing technology applied to online detection of pharmaceutical industry [j]. *Internet of Things Technologies*, 8:034, 2011.

[18] A. Karargyris, O. Karargyris, and A. Pantelopoulos. Derma/care: An advanced image-processing mobile application for monitoring skin cancer. In *IEEE ICTAI*, volume 2, pages 1–7, 2012.

[19] I. Krasin, T. Duerig, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2016.

[20] H. Lee, A. F. Smeaton, N. E. OConnor, G. Jones, M. Blighe, D. Byrne, A. Doherty, and C. Gurrin. Constructing a sensecam visual diary as a media process. *Multimedia Systems*, 14(6):341–349, 2008.

[21] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, Feb. 2006.

[22] S. Lin, Özsu, et al. An extendible hash for multi-precision similarity querying of image databases. In *VLDB*, pages 221–230, 2001.

[23] D. S. Linthicum. *Enterprise Application Integration*. Addison-Wesley, 2000.

[24] H. Manzoor, Y. SinghRandhawa, and E. D. G. Amritsar. Comparative studies of algorithms using digital image processing in drug industry. *IJSRP*, page 418, 2014.

[25] A. Michaels et al. Vision-based high-speed manipulation for robotic ultra-precise weed control. In *IEEE/RSJ IROS*, pages 5498–5505, 2015.

[26] M. Nagle et al. Non-destructive mango quality assessment using image processing: Inexpensive innovation for the fruit handling industry. In *Conference on International Research on Food Security, Natural Resource Management and Rural Development*, pages 1–4, 2012.

[27] L. Peng, K. S. Candan, C. Mayer, K. S. Chatha, and K. D. Ryu. Optimization of media processing workflows with adaptive operator behaviors. *Multimedia Tools and Applications*, 33(3):245–272, 2007.

[28] D. Ritter et al. Hardware accelerated application integration processing: Industry paper. In *DEBS*, pages 215–226, 2017.

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

[30] D. Ritter, N. May, K. Sachs, and S. Rinderle-Ma. Benchmarking integration pattern implementations. In *DEBS*, pages 125–136, 2016.

[31] SAP SE. SAP Social Intelligence Data Harvesting from Social Media Channels. https://goo.gl/Awi78v, 2017.

[32] L. F. Sikos and D. M. Powers. Knowledge-driven video information retrieval with lod: from semi-structured to structured video metadata. In *ESAIR*, pages 35–37, 2015.

[33] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *IEEE ICCV*, pages 1–8, 2007.

[34] K. R. Stokke, H. K. Stensland, C. Griwodz, and P. Halvorsen. Energy efficient continuous multimedia processing using the tegra k1 mobile soc. In *ACM MoVid*, pages 15–16. ACM, 2015.

[35] Y. Taigman et al. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014.

[36] R. Tillett. Image analysis for agricultural processes: a review of potential opportunities. *Journal of agricultural Engineering research*, 50:247–258, 1991.

[37] C. Tzelepis, Z. Ma, V. Mezaris, B. Ionescu, I. Kompatsiaris, G. Boato, N. Sebe, and S. Yan. Event-based media processing and analysis: A survey of the literature. *Image and Vision Computing*, 53:3–19, 2016.

[38] S. Vemula and C. Crick. Hadoop image processing framework. In *IEEE International Congress on Big Data*, pages 506–513, June 2015.

[39] J. A. Watlington and V. M. Bove. A system for parallel media processing. *Parallel Computing*, 23(12):1793–1809, 1997.

[40] P. I. Wilson and J. Fernandez. Facial feature detection using haar classifiers. *J. Comput. Sci. Coll.*, 21(4):127–133, Apr. 2006.

[41] C. Yang et al. Recognition of weeds with image processing and their use with fuzzy logic for precision farming. *Canadian Agricultural Engineering*, 42(4):195–200, 2000.

[42] H. Zhu, Z. Shen, L. Shang, and X. Zhang. Parallel image texture feature extraction under hadoop cloud platform. In *ICIC*, pages 459–465, 2014.