Structured Email (sml) BoF

IETF 116, Yokohama, JP

List: sml@ietf.org

Structured Email: Background and problem statement

Presenter: Hans-Jörg Happel
happel@audriga.com

IETF 116, Yokohama, JP

Preface: Knowledge representation on the web

Technology

- **Semantic Web:** make data on the internet machine-readable
- **RDF:** W3C standard
  - General method for description and exchange of graph data
  - Can be stored in files or inline HTML
  - Serializations/media types: application/rdf+xml, text/turtle, ….
- **Schema.org:** Shared vocabulary
  - 797 Types (Movie, Person, Restaurant), 1457 Properties
  - Used by > 10 Mio sites

Applications

- **Search:** Schema.org annotations in Websites render rich search result snippets in major search engines
- **Sharing and Embedding:** Facebook Open Graph, Twitter Cards
- **Wikidata:** provide machine-readable version of knowledge in Wikipedia

What is “structured email”?

- Email (content) which is not for manual human processing, but for semi- or fully-automated interaction

- Why?
  - Automation: lots of email is transactional and structured data could ease processing
  - Data sovereignty: Email is unique in bridging private and public information space

Structured email examples

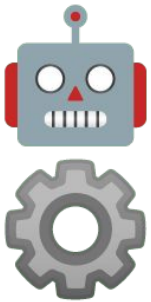
- **RFC-based**; for particular use cases (e.g., Calendar invites/iMIP, ARF …)
- **Vendor-specific** “mail-in”-APIs (e.g., Majordomo Mailinglist “subscribe”; Helpdesk/ticket systems)
- **Domain-specific** applications (e.g., Chat over IMAP/DeltaChat)
- **Generic** approaches (e.g., Schema.org annotations added to email content; started in 2013 by Gmail)

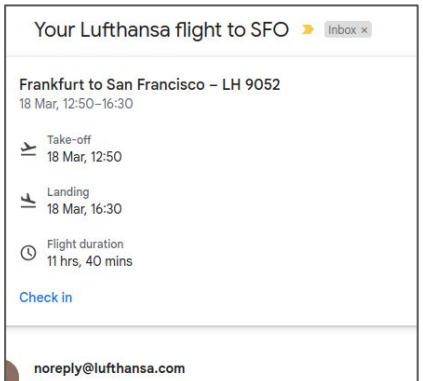
“Schema.org for email” (aka “Email markup”)

Sender

+

Receiver





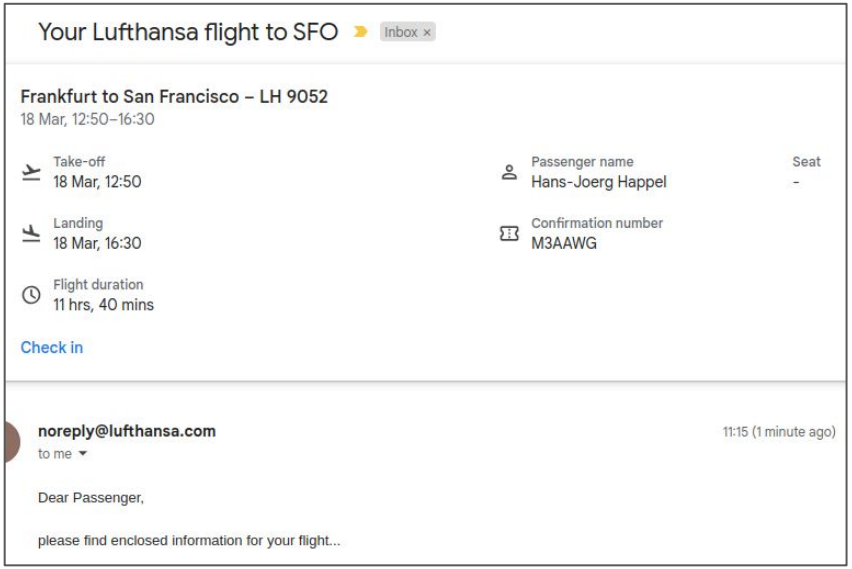
. . .

“Schema.org for email” (aka “Email markup”) (detail)

Senders add in a <script> tag in email text/html body part (downwards compatible):

```
{
  "@context": "http://schema.org",
  "@type": "FlightReservation",
  "reservationNumber": "M3AAWG",
  "reservationStatus": "http://schema.org/Confirmed",
  "underName": {
    "@type": "Person",
    "name": "Hans-Joerg Happel"
  },
  "reservationFor": {
    "@type": "Flight",
    "flightNumber": "9052",
    "airline": {
      "@type": "Airline",
      "name": "Lufthansa",
      "iataCode": "LH"
    },
    "departureAirport": {
```

Email UIs show:



Schema.org for email: Current adoption

Sender side

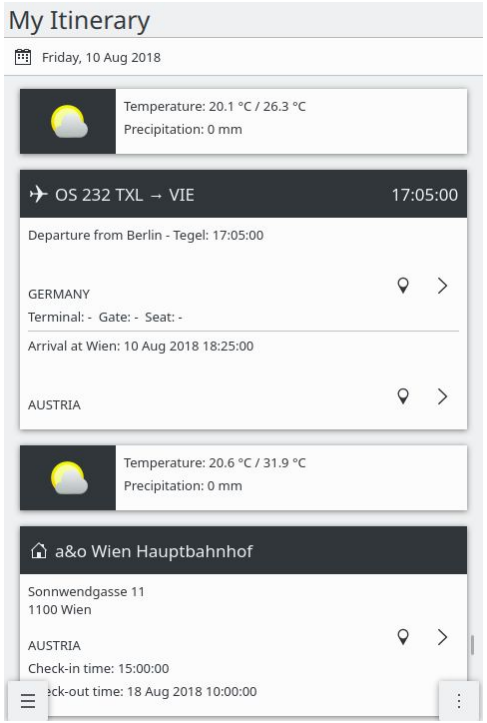
- Some ESPs and larger brands (Asos, Etsy, Google Play)
  - Display requires registration and approval for each Freemail provider
- Complementary data extraction
  - Freemail-provider specific processing
  - Open Source tools (KItinerary)

Receiver side

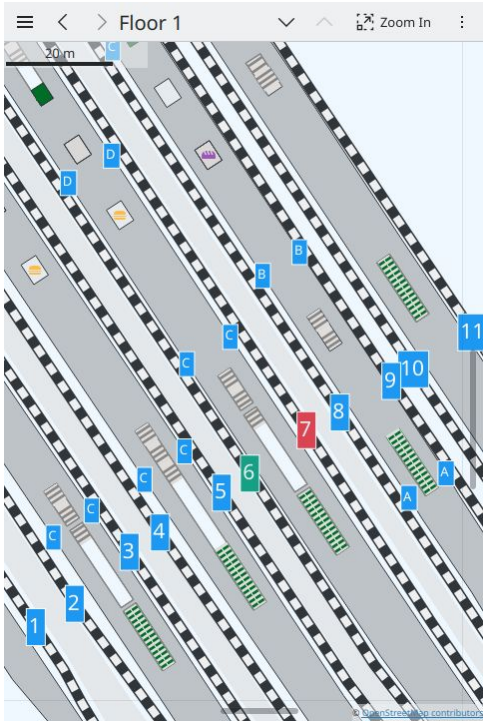
- Large freemail vendors covering a large part of the market
  - Gmail
  - Verizon Media (AOL/Yahoo)
  - 1&1 (WEB.DE/GMX)
  - Zoho
- Open Source tools (experimental)
  - KMail
  - Nextcloud Mail
  - KDE Itinerary

Schema.org for email: KDE Itinerary app

Open source app which provides travel itineraries **based on data extracted from emails**



The app can also provide **additional contextual info** (here: station map including live elevator status)



Schema.org for email: Current issues

- Current usage is one-way only (sender to consumer)
  - In particular: large senders to big freemail providers
- Interoperability (Freemail providers use partly different markup)
- Complex onboarding
  - Manual registration
  - Sender requirements
  - Difficult to test
- Senders need more confidence in client-side usage

Side note I: What about MIME types?

- MIME types are mainly “file/artifact-oriented”
- MIME types for RDF data exist (application/rdf+xml, application+json, …)
- Breaking down structured data into further MIME types seems impractical
  - Plethora of required MIME types (“application/flightreservation”, …) and body parts
  - Does not play well with relations between entities (FlightReservation → Airline)

- In essence
  - Structured data is orthogonal to MIME types
  - While MIME types for structured data exist, structured email is about making it a “first order citizen” for MIME messages/email clients, beyond a “mere file attachment”

Side note II: Where do vocabularies come from?

- How do sender and clients agree on vocabularies?
- Modeling extensive vocabularies (what is a “restaurant” etc) is not a goal of this BoF
- So where would vocabularies come from?
  - Already established vocabularies (e.g., Schema.org, Wikidata, …)
  - In special cases: from particular RFCs (e.g., pEp, VacationNotice)
  - Vendor specific (c.f. MIME types → you can send arbitrary attachments, transparent to your email client)
  - Discovery mechanisms?

Why this BoF?

- **This is a relevant extension to email**
  - It is already useful for itself in its current form (→ following talk)
  - It provides a mechanism that can be reused in other work (→ e.g. pEP, VacationNotice)
    - Prior RFCs (iMIP, MDNs, …) do not use common interaction mechanisms → complicates implementation in email clients
  - Foundation for future use cases (e.g., end users sending structured email; interactive, dynamic email)
  - **Overall, email needs to be enabled to remain relevant in the future**

- **There is work required** to go beyond current “big senders → big providers” usage
  - Several aspects are loosely defined; particularly difficult to adopt in the long tail
  - Issues identified stem mostly from implementation experience
  - Current usage model is unidirectional (end users are not supposed to send structured email)
  - Extensions are possible while retaining compatibility to existing usage

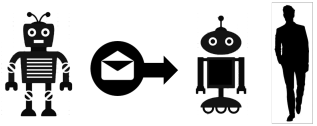
Structured Email: ISP experiences with Schema.org in email

Presenter: Conny Junghans
conny.junghans@1und1.de

IETF 116, Yokohama, JP

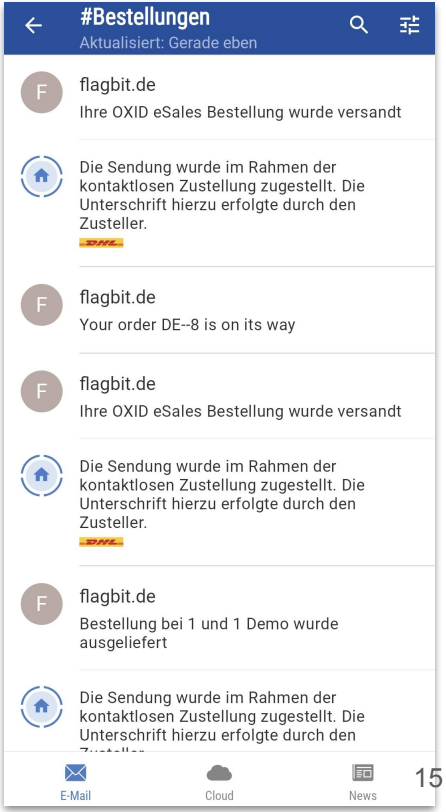
ISP experiences with schema.org in email - Status Quo

- Why did we get into schema.org in the first place?



  - Ever-increasing B2C communication, mostly machine-generated
  - Provide most important information at a glance

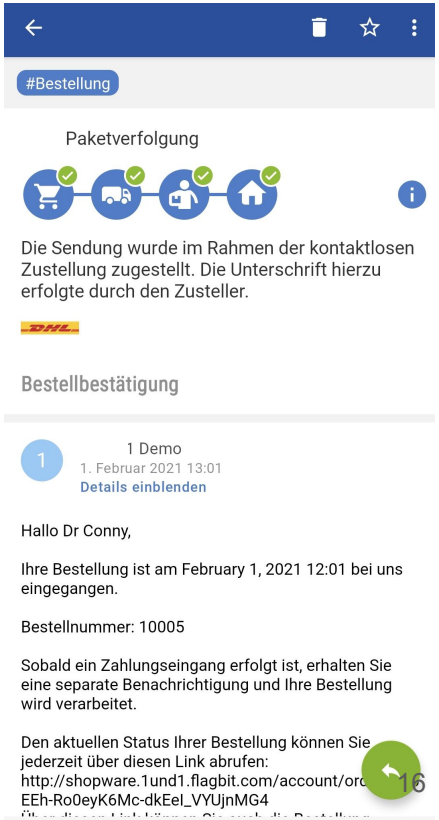
- Currently, mix of ML and schema.org (using gmail email markup)
  - Structured data gives control for the “email summary” to senders



ISP experiences with schema.org in email - Status Quo

- We show updates to the content of (shopping) emails that are obtained from related shopping emails or third-party APIs

- We provide modules for 4 major shop systems that automatically add schema.org to all transaction emails
  - Ease of adoption for smaller shops
  - Get general adoption rate up, then hope for FOMO on sender side



ISP experiences - adoption hurdles

- Knowledge and Incentive
  - Senders need to know that structured data can be added to emails and need to have a big enough incentive to do so -> ISP-specific solutions hurt here
- Documentation and Testing
  - Which mailbox providers / clients support schema.org in emails?
  - Which entities are supported?
  - What will the email with schema.org look like in the client?
  - How can I test my setup before going live?
- Mail Security
  - Spammers will be the first to adopt any way to highlight emails in mailboxes
  - For everything beyond “show the email as is” we need to take responsibility to keep our users save

Structured Email: Clarifying questions and open discussion

IETF 116, Yokohama, JP

Structured Email: Issues for standardization

Presenter: Hans-Jörg Happel
happel@audriga.com

IETF 116, Yokohama, JP

Suggested standardization work

- Focus on “Schema.org for email” baseline
  - Overall approach
  - Internet Message format
  - Trust
- Ensure compatibility with existing implementations
- Potential later topics
  - Empowering users to send structured email
  - Discovery aspects
  - “Dynamic” email (interrelation with other approaches, incl. AMP Email/Actionable messages)

Schema.org for emails: Standardization issues

- Partial vs. full representation
- Structured data formats
- Structured data vocabulary
- Relating structured data to message content
- Addressing multiple MUAs
- Message updates
- Responding (to actions)
- Efficient processing
- Data extraction

Note: issues and proposals are preliminary and incomplete / intended for illustration

Issue: Partial vs full representation

- Status quo
  - Structured data is embedded in text/html body part (see previous example) → partial representation of content
  - Issue: proper usage and display of regular text/html body cannot be prescribed (e.g., complementary vs fallback)

- Proposal
  - Allow full representation by adding multipart/alternative text/json+ld
  - Perhaps allow partial content fallback (e.g., multipart/alternative text/rdf+html)

Issue: Structured data formats

- Status quo
  - Structured data can be expressed as JSON-LD or HTML microdata
  - Issue: receivers need to support multiple formats

- Proposal
  - Keep options based on rationale to make it easy for senders in the first place
    - E.g., only certain options might be practical to adopt
  - Also allow application/rdf+xml / text/turtle representations of RDF?

Issue: Structured data vocabulary

- Status quo
  - Schema.org for email currently uses just a fraction of the Schema.org vocabulary
  - Issues:
    - Very limited use cases
    - No extension mechanism

- Proposal
  - Allow for general RDF, probably consider a dynamic registry mechanism at later time (rather than static registry as in case of media types)
  - Consider specification facility for structured data required in certain RFCs (e.g., pEp, VacationNotice)

Issue: Relating structured data to message content

- Status quo
  - Structured data is embedded in the text/html body part
  - Issues:
    - Scope of structured data in unclear (describing full message? text/html content? Other body parts?)
    - No fine-grained cross-referencing of HTML content (except for HTML microdata case)

- Proposal
  - Options
    - Allow for multipart/related nesting of structured data? → too complex?
    - Allow for a CID/MID-like reference mechanism?
    - Reversely, allow to reference HTML element ids and body parts from structured data?

Issue: Allow clients to hide emails and email body parts

- Status quo
  - Email clients will typically show all emails and all body parts to the end users
  - Issues:
    - “Technical” emails get normal users confused
    - “Technical” body parts get normal users confused
  - Related issues outside SML: encryption keys/signatures, inline images, …?

- Proposal
  - Add message header signaling message is meant for automated processing
  - Add body part header for hiding body parts meant for automated processing?
  - Even allow to advertise extensions (e.g. PGP plugin)

Issue: Addressing multiple MUAs

- Status quo
  - Different MUAs (email clients) of a user can coexist without knowing each other
  - Issues:
    - How to avoid resp. prescribe “multi-processing” of structured data
    - How to address a particular MUA if needed
  - Related issues outside SML: calendar invites, filters, throttling (!), encryption (pEp)

- Proposal
  - For multi-processing: Use IMAP FLAGS? (similar to MDNSent?)
  - For addressing: Client-Id? Header field?

Issue: Message updates

- Status quo
  - Structured data cannot be updated/revoked
  - Issues: No clear path to update/revoke data
  - Related issues outside SML: “Status update flooding”, “Recalling” messages, recalling encryption keys (pEp)

- Proposal
  - Options:
    - Introduce “REPLACES” header, referencing MESSAGE-ID? (exists?)
    - Solve within structured data (similar to iMIP updates)

Issue: Responding (to actions)

- Status quo
  - ConfirmActions can be confirmed by a HTTP POST request
  - Issues:
    - No way to reject
    - No way to track “responded “state
    - No audit record for “response”
    - Email clients might not want to support HTTP requests

- Proposal
  - Allow for confirmation using a structured email response (similar to iMIP responses, DSNs)
  - Introduce IMAP FLAGs (similar to MDNSent/hasAttachment)

Issue: Efficient processing

- Status quo
  - Email body needs to be analyzed for structured data
  - Issue: costly/impossible in cases when only email headers are fetched
  - Related issues outside SML: attachment processing, iMIP?

- Proposal
  - Options
    - Use IMAP FLAGS (similar to hasAttachment)
    - Introduce header field(s)

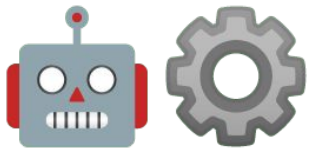
Issue: Data extraction

- Status quo
  - Some vendors/tools apply data extraction techniques to emails which do not contained Schema.org for email (even if their content could be described by it)
  - Issues
    - Extracted structured data is stored in proprietary way
    - Might not be available to all clients

- Proposal
  - Update email body content
  - Consider portable storage format for email metata?
  - Probably related: IMAP ANNOTATE (RFC 5257)

Discussion / Next steps

- Structured email is a relevant extension to email
- Work needs to be done to make this more broadly available and applicable



List: sml@ietf.org