MENAID

MENAID - Methods and tools for next generation document engineering
2011-2014, National Science Center,
grant no. 2011/01/B/ST6/06500
Website: menaid.org.pl

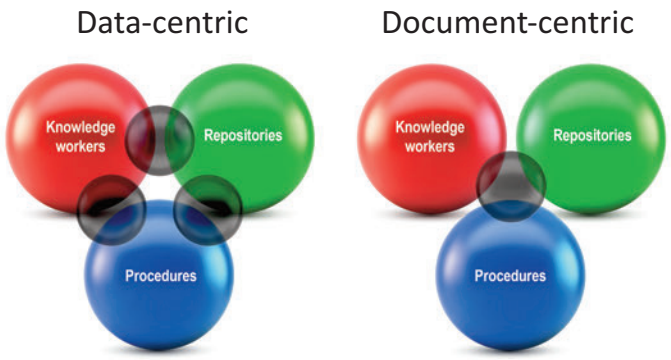
Interactive documents

for network organisations

GDAŃSK UNIVERSITY OF TECHNOLOGY

Document centric collaboration

The MENAID project researches new document architectures to reduce excessive e-mail messaging, save time of knowledge workers, reduce operational costs of knowledge organisations and rationalise their information flow.



Network organisations provide knowledge workers collaborating on reaching a common goal with some computer-mediated communication facility, linking them across boundaries to interact and exchange information. The most popular is email – combining simple textual messaging with a possibility to disseminate attachments with just one click to one or multiple workers, and natural support for asynchronous work – enabling collaborators to respond any time after receiving a message and to copy or archive messages for further use. Attachments enable collaborators to exchange documents in any format, and practically of any size (if a document exceeds a size limit its URL may be sent instead).

Intellectual resources of the organisation include in general:

- **Workers**, who bring to the organisation their tacit knowledge, usually not recorded anywhere;
- **Procedures**, which represent implicit knowledge, enforced in the organisation by existing standards, legal regulations, or just gathered by the organisation through the experience as 'best practices';
- **Repositories**, with explicit knowledge encoded and organised in a form enabling effective search, categorisation and retrieval of data, such as databases, digital libraries or knowledge bases.

Workers are the key intellectual resource of the organisation for their ability to resolve decision problems that cannot be automated easily (if at all), therefore they must be provided with interaction mechanisms enabling them to access and share all three types of knowledge available across the organisation. Interfaces enabling that must provide both effective communication, broadly understood as the exchange of information units, and coordination of activities performed by collaborators.

There are two paradigms for implementing such interfaces, commonly referred to as data-centric and document-centric. The former requires that during interaction any two collaborating parties use identical data structures to exchange the information, while the latter assumes documents to be simultaneously information and interaction units. This distinction is outlined in the figure above.

A data centric processing paradigm in the organisation necessitates implementation of many specialised interfaces. The most common one is between workers and repositories; it may involve queries, performed directly in some specialised query language, or indirectly with forms – documents with a strictly defined syntax to be parsable by computers but comprehensible to people. If procedures of the organisation are automated – for example implemented as workflow processes, they may also interact directly with repositories using some kind of syntactically correct service calls. Finally, individual activities of workflow processes may also provide various interfaces to serve as an intermediary between collaborating knowledge workers and the procedures. Unfortunately, if the principal communication facility is email, knowledge workers must implement procedures of the organisation manually. They must know, and (hopefully) understand them. In doing so they must not make mistakes, since they cannot be controlled in what they send and where, and have a limited capability to distinguish and trace their concurrent activities. Contrary to that, document centric processing paradigm helps to introduce a universal interface between the workers, procedures and repositories, with proactive documents providing collaborators with information (document content) and interaction facility (document functionality). If such documents can be freely disseminated between workers and services of the organisation no other interface would be necessary.

New document architectures

Proactive documents researched in the MENAID project are capable of initiating activities independently of any external system or user and can provide a uniform interface between resources of any network organisation implementing a document centric processing paradigm. In particular they are:

- **Mobile**, i.e., can migrate on their own between users and services of the organisation using standard services provided by the Web;
- **Executable**, i.e., enable exploration of information resources related to their content via interaction of users with logical structure elements;
- **Persistent**, i.e., are technology neutral, forward compatible and not subscribing to any particular vendor, tool or format.

Mobility

Each document has an embedded description of its migration path, which describes activities and transitions that must be followed during its lifetime. An activity represents a piece of work to be performed by the worker on the incoming document content, whereas a transition indicates where the outgoing document (or documents), constituting a result of the just completed activity, should migrate next. This idea is outlined in the figure below, where activities are represented by boxes, transitions by arrows, and two distinguished start and termination activities by filled and, respectively, outlined circles.

Workers may interact with document content using any device currently at their disposal, from workstations to laptops, to tablets or smartphones. A specially designed Local Workflow Engine (LWE) client is installed on each device; it can unpack and pack the documents and send them to other workers of the organisation.

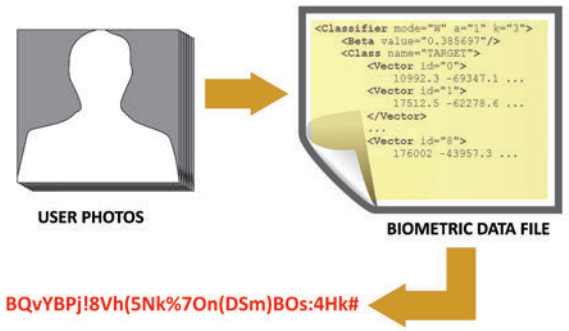
Executability

Functionality of LWE clients is generic and limited to handle several canonical document transition patterns, sufficient to implement any process in the organisation. Any specific functionality needed to handle a given content is brought to the worker by the document itself. The role of LWE is to enable this functionality on the execution device by activating appropriate document services. We distinguish three types of such services:

- **Embedded**, which are pieces of code or scripts that may be executed directly on the receiving worker's device;
- **Local**, implemented as scripts that can test for availability and eventually activate various tools installed on the worker's device;
- **External**, specifying services provided by some remote server, which the arriving document would like to call via the receiving worker's device.

Services enabled us to introduce a couple of novel features to documents. One has been biometric protection of the document content, and another augmenting them with the negotiation capability.

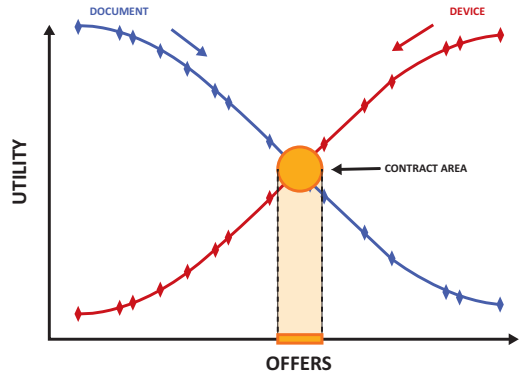
Biometric protection exploits encryption mechanisms supported by existing document formats but relieves users of inventing and memorising passwords, owing to the specialised embedded functionality for automatic password generation based on the user's biometric facial data, as shown below.



Preparing a biometrically protected document starts at some registration point of the organisation with a service generating a facial data file based on a set of photographs of the worker to whom the document will be sent. The file is put to another specialised service to generate a password, used next to encrypt each document being sent to the registered worker with a standard encryption tool provided for its format. Currently we support PDF and ZIP. The encrypted document is sent next

to the registered worker along with his/her facial data file. When attempting to open the received document its embedded biometric service takes the recipient's photo, verifies his/her identity against the received facial data file, and either rejects the worker or generates a password opening a document based on the positively verified facial data. Our biometric protection mechanism is as strong as possible, i.e., depends only on the strength of specific encryption algorithms used by available tools intended for the particular document format.

Negotiation capability of a document is required to resolve potential conflicts between its services and devices providing them with a concrete execution context. These contexts may change, depending on what personal device is currently used by the worker and what is his/her actual location. We have adopted a simple bargaining model to negotiate a contract between a document and the execution device by exchanging offers and counter-offers. An offer is a set of items, each one representing a concrete value of some specific attribute of service execution. All possible offers constitute a bargaining set in which a contract satisfying both parties is sought. One offer may provide a different utility to each party , so it is expected that the contract will be an offer providing possibly the highest utility that is simultaneously acceptable to both parties. A party does not know preferences of its opponent, so it makes offers starting from the one providing it the highest utility. This scheme is outlined schematically in the figure below.



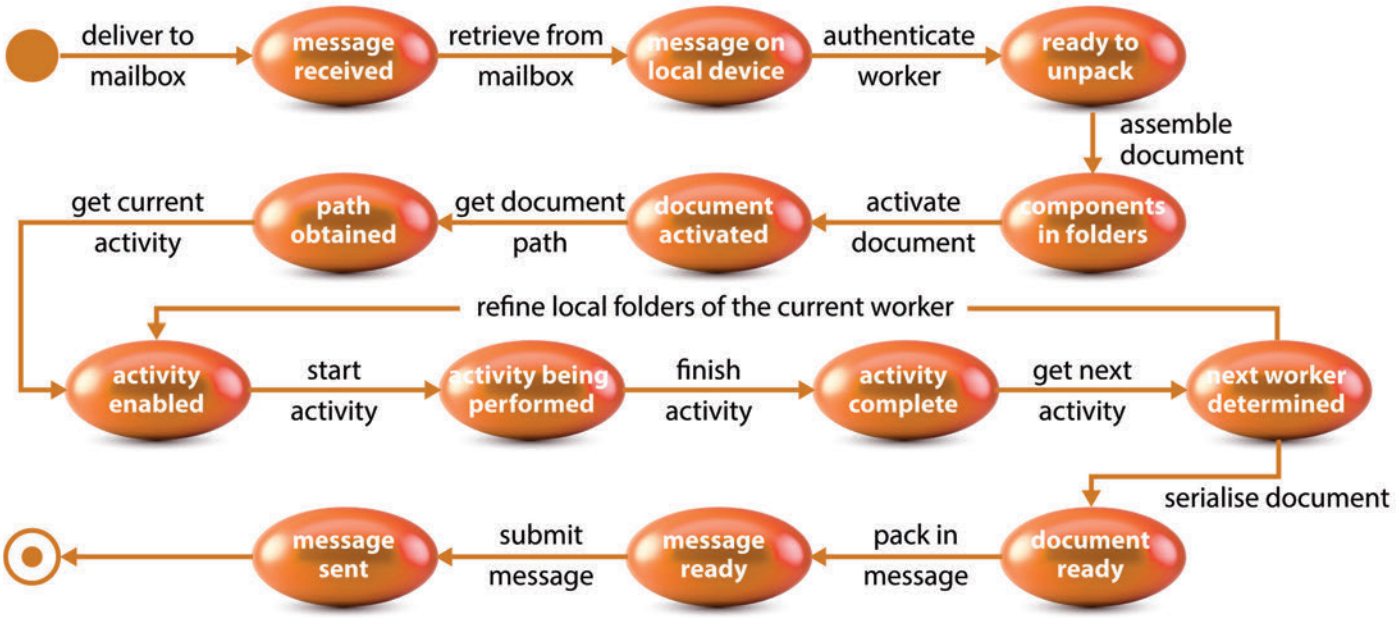
If the range of possible values for each attribute is wide the bargaining set may be arbitrarily large, making sequences of offers and counter-offers exchanged during negotiations excessively long. In order to speed up the negotiation process we have augmented proactive documents with a machine learning capability. With that kind of embedded functionality, our documents can learn to recognise preferences of an execution device after a few first encounters and guess a contract faster during encounters with it in the future.

Persistence

Mobile and executable documents must be able to cope gracefully with unknown future data formats, if solutions they provide are to persist in the organisation for a realistically long time. Design rationale for the proactive document architecture developed in the MENAID project has been accompanying document data by a standard description enabling the receiving worker's device to pick up an appropriate mechanism making the data usable to the knowledge worker, augmented if necessary with a minimal document functionality. Document data description, and related mechanisms to handle them, are based on the MIME standard, used by email systems to cope with encoding transformations of native representations of content – subject to encoding transformations when passing through mail transport protocols with various data limitations. When combined with embedded functionality brought to the receiving device by the document, the receiving LWE client can adapt to its content in the same fashion as functionality of web browsers can be expanded with plugins. By implementing our proactive documents on top of a regular email system we have been able to bring to network organisations all the benefits of multi-agent systems, such as self-organisation and self-steering in implementing arbitrary complex collective behaviors of workers with simple individual agent behaviors, without a need to implement a full-size agent platform which may quickly fall into obsolescence.

Smart email

A document can enable interaction between its content and the worker as long as it remains active on the workers device. This is the activity phase of a mobile document, which starts when it arrives to some worker and ends when the document is ready to be sent to the next worker. During its transition from one worker to another a document is just an attachment to the email message. In order to handle switching documents between the activity and transition phases our LWE client has been implemented as a lightweight email client, installed on each worker's device as a standalone application or a plugin to the existing email client. Its generic functionality is to retrieve serialised document attached to the messages in the worker's mailbox, deserialise them an activate their functionality, and when the current activity is to be concluded, serialise them and pack into a message submitted to the worker's email server for sending to the next worker. A state diagram of the LWE client is shown in the figure below.



The initial state of LWE is when the message is placed in the respective worker's mailbox hosted by some email server. Depending on the particular collaboration pattern, transition to the next state may require delivery of one or more messages to the mailbox. LWE identifies in that effect all relevant messages in the mailbox and retrieves them to a local inbox at the knowledge worker's device. Making document components ready to unpack may require authentication of the worker, for example with the biometric mechanism described before. Unpacking of the retrieved messages enables assembling the document, which involves creation of the local folder structure to keep document components. Activation of the document enables its embedded functionality, so it may interact with LWE, the worker, a local system of the execution device, and if specified, services of external servers. This interaction begins with determining the activity to be started based on data indicated by the document migration path. Based on that a thread of the related workflow process is resumed properly, i.e. the right activity is enabled each time document components packed in the message migrate from one worker to another. The enabled activity is started and performed using the document services: automatically, or via interaction with the worker. When the required work is done the current activity is considered complete. This implies determining the next activity to be performed and the respective worker responsible for that. If the worker is the same, the content of local folders is refined to prepare document components processed so far for the next activity and enabled for the next cycle. Otherwise, document components are serialised, depending on the particular collaboration pattern packed in one or more messages, submitted to the email server and sent out.

The login page allows a worker to specify what mailbox to open to get messages with documents to work on with his/her email address and password. At this phase a network connection is needed to retrieve all documents addressed to the worker from the indicated email server. Worker's credentials provide also a basis for standard authentication. The todo list indicates all processes the documents retrieved to the worker's local inbox are involved in. They may be sorted with regard to their urgency, calculated for each thread as the amount of time remaining to complete the current activity indicated in the migration path, date of arrival of the first document involved in the process thread, or the process thread ID. By selecting a desired thread its relevant activity page is opened. Depending on the particular collaboration pattern, starting the current activity may require receiving one or more document components. It may be seen in the activity page that five document components belonging to thread 'MIND0002' are expected. Upon receiving all of them the 'Start' button becomes active and the worker may start to perform it.

Example screenshots illustrating the LWE client at work.

From this point, network connection is not needed if the document components indicated to the LWE client declare no interest in accessing any external service. It may also happen that the execution device does not allow the incoming document to connect to the currently available network, which may be not secure or too costly to access at the moment – so the document must either wait until the worker changes its location and gets access to another network, or negotiate with the device an alternative solution. Performing the activity implies calling the respective service by the proactive document to interact with the worker. In the example above, a spreadsheet tool available locally at the worker's device has been called.

The dark side of email messaging is email overload, that has become a serious economic problem for network organisations today. Feelings of mail overload and strain reported by workers have its source in the communication of content, when an excessive volume of messages must be processed in a given period of time by a single worker, as well as in the coordination of activities, when a worker is delayed by waiting for responses related to a specific task, or is interrupted (distracted) by messages related to other tasks. Negative effects of communication and coordination are often combined, as workers in network organisations tend to use e-mail for ad hoc task management. Ad hoc means that no specific collaboration patterns are used, as no email systems yet can provide a sufficient support for them. Task management involves multitasking, which is not natural for the human mind when forced to take over many activities at once. In consequence, manual implementation of pre-emption (interrupting and resuming of tasks) by the worker is perceived as tedious and exhaustive, often resulting in reading messages in different order they arrive or even worse, not reading some emails at all!

Proactive document architectures developed in the MENAID project can make email systems smarter, without a need to invest in new technologies or introducing revolutionary changes in the way workers of the network organisation cooperate. All that our architectures can do is to contribute to a significant reduction of the organisation's operational costs and rationalising its information flow.

Communication of content

Although MIME types, commonly recognised by email systems, can effectively identify to the receiving worker what content has been attached, performing an activity on it by the recipient may require additional arrangements. One is when it becomes

necessary to make an a priori agreement with the sending worker on what tools shall be used for the identified (standard) content. Another is finding tools for the content type identified as non-standard.

Standard content type may be handled by popular tools in different ways, despite of commonly agreed document formats. Examples include formatting of tables with merged cells, page formatting due to font substitutions by word processing tools, etc. Differences of this kind are irrelevant to the organisation providing its workers with systems and tools of the same make and version, putting them in some confined space and requiring to perform their activities during office hours. But this assumption is no longer realistic in a dynamic virtual organisation setting, where knowledge workers are spread over a large area, change their locations frequently (either on business travel or just being between office and home), work in over-time, and use different (most often their personal) execution devices. Handling of this problem by the proactive document, with embedded functionality dedicated to the particular type of content, or at least able to adapt the document content to the local device specificity, seems to be more appropriate than forcing a monoculture of document processing tools across the organisation.

Non-standard content type, unknown to MIME, is indicated with a generic application/octet-stream identifier, so the receiving email client has no hint on what tool to use to open the document. If never dealt with such a content before, the receiving worker may reasonably expect instructions attached to the document on what tool to use to perform the current activity. This situation may involve even a series of messages to negotiate a solution if conflicts arise. Sometimes email users may agree on exchanging private content types, for which MIME provides identifiers with a special prefix. Since not-registered and non-standard there is no guarantee that they will not be replaced by something else in the future. Unfortunately, a non-standard content necessitates additional adjustments of local systems, despite of reaching an agreement on a document format by collaborating knowledge workers. Configuration of their systems often cannot be automated if based only on the received content type descriptions – so upon receiving documents, further consultations (most likely by exchanging additional mails) may be required. The need for that, however, may be eliminated by properly configured document services brought with its related content to the execution device.

Example screenshots illustrating the LWE client at work.

Coordination of activities

In a mobile interactive document system individual workers perform activities on documents independently, using their physically separated personal devices, and yet collaborate on achieving a common goal. It is possible owing to the migration path embedded in each proactive document that they receive for processing. This path defines for each document a workflow process, which combines activities and transitions forming specific document flow patterns; they implicitly involve workers in the collaboration process and provide process-wide coordination of their respective activities. Owing to that, implementation of business procedures in a network organisation may be significantly simplified and eliminated from the everyday routine of knowledge workers, overwhelmed with mass influx of emails. It has been shown by researchers of the Workflow Patterns Initiative, a joint effort of Eindhoven University of Technology and Queensland University of Technology started in 1999, that arbitrary complex workflow processes in real-life organisations may be composed of elements forming a relatively small and well defined set of collaboration patterns. Based on that development, the MENAID project has worked out a subset of control-flow patterns that can be implemented with common email services. As a result three categories of document-flow patterns have been identified: distributed state patterns, with transitions depending only on the state of a single activity performed by the worker at some location in the system, coupled state patterns, with transitions depending on states of two or more activities performed simultaneously by two or more workers at separate locations in the system, and embedded state patterns with block activities implemented as sub-flows.

Distributed state patterns describe situations in which the next activity to be performed can be determined solely on the state of the current activity. We distinguish four such coordination patterns:

- **Sequencer**, when after completing an activity a knowledge worker sends to another worker one or more resulting documents in a sequence;

- **Splitter**, when each resulting document is sent to a different worker. Splitter may be cloning, i.e., the resulting document is multiplied by making identical copies before sending, or decomposing, i.e. the resulting document is split in separate fragments before sending;

- **Merger**, which complements the document splitter pattern, may involve various document functionality, depending on whether the preceding splitter has been cloning or decomposing. It may be as simple as the concatenation of chunks of text, or quite complex as synchronising documents with sophisticated content merging algorithms. In either case the respective functionality enabling that is brought by the arriving document components to the receiving worker's device;

- **Iterator**, which enables repeated execution of the sequencer pattern controlled by a condition specified in the respective document migration path. Functionality brought by the arriving document to the respective worker's device can determine loop termination based on the current document content.

Coupled state patterns deal with the situation when completion of the activity performed by one worker may require notification on a state of some activity performed by another worker of the organisation. Collaboration patterns enabling that involve the notion of asynchronous signals, sent between different parts of the workflow process. Such a signal may be implemented in a network organisation in many ways, e.g. with texting, phone calls, instant messaging, or just 'out-of-bound' email messages. We distinguish three collaboration patterns of this kind:

- **Deferred choice**, when sending a given document has to be postponed until the worker gets information to whom it should be sent;

- **Milestone**, a simplified version of deferred choice, which only purpose is to block some activity of one worker by another;

- **Cancel activity**, performed by the worker sending a special cancellation signal to another worker. A worker receiving the signal deletes the document he/she has been working on, or prepares for it deletion if it has not been delivered yet. If the document has been already processed and sent out the cancelation signal is ignored.

Embedded state patterns involve a worker performing a single activity by delegating to someone else a subprocess with activities not specified originally in the migration path of the arriving document. We distinguish two such patterns:

- **Internal subflow**, when the authorised performer of the current activity adds new activities and assigns workers to perform them by editing the original migration path embedded in the received document;

- **External subflow**, when the performer of the current activity calls some external service which neither structure nor identity of thus 'subcontracted' workers are known to the current activity performer. Separation between the main process and the external subprocess is provided by some intermediary server, to which a document is uploaded for processing, and downloaded when completed; in between, the subcontracted external process downloads it for processing and upon completion, uploads back to the intermediary server. Upload and download service calls are assumed to be blocking, i.e., the current (contracting) worker and his/her unknown contractors wait until the respective service call has finished. Note that in such a case the external process may be implemented in any technology, not necessarily as a document centric application.

Towards knowledge-based

Proactive email attachments can add to the email metaphor a process perspective. It allowed us to address five major challenges to email as the enabling technology for knowledge based organisations, where procedures constitute an important knowledge resource, as indicated at the beginning of this paper:

- **Tracing** of concurrent activities for proper distribution and selection of work, as well as for acquisition and synchronisation of information from external sources;

- **Controlling** importance of activities implied by the received document content in terms of the required effort and time to complete them;

- **Management** of activities extended in time and monitoring tasks they belong to, especially with regard to causality analysis of the related events;

- **Combining** received messages in order to file all documents related to the particular activity;

- **Overview** of the current activity in the context of task execution and its resources.

Prof. Bogdan Wiszniewski, Head
Department of Intelligent Interactive Systems
Faculty of Electronics, Telecommunications and Informatics,
Gdansk University of Technology
ul. Narutowicza 11/12, 80-233 Gdansk, Poland
Tel: + 48 58 347-1089 | Fax: +48 58 347-2222 | Email: bowisz@eti.pg.gda.pl

GDAŃSK UNIVERSITY
OF TECHNOLOGY