How Physical Objects And Business Workflows Can Be Correlated

Andreas Wombacher

Database Group, University of Twente, Enschede, The Netherlands

Email: a.wombacher@utwente.nl

Abstract—Service Oriented Architectures ease integration of heterogeneous systems, such as sensor data and workflow systems. Systems are integrated since they model an overlapping part of the physical world, i.e., physical objects exchanged between different parties. For workflows handling physical objects, the correlation of sensor data with workflow states and workflow state changes are investigated in this paper. Further, the implications of the state or state change correlation on the workflow execution and the support by existing service infrastructures is discussed.

I. INTRODUCTION

In case a workflow describes the handling of physical objects, workflow systems and sensor data are more and more integrated. An example is context aware applications, where workflows use context information supporting users in fulfilling a task in a physical environment [1]. Another example is logistics processes, where sensor information facilitates fleet management and package tracking applications [2]. The SOA support for integration of workflow systems and sensor data motivates an investigation of tight and loose integration or correlation of workflow systems and sensor data and the implications on workflow properties.

Sensor data as well as workflow systems describe a view on the physical world consisting of physical objects with properties and a location. This view can never be complete and is not always precise. Sensor data introducing physical objects into the digital world, for instance, are potentially imprecise [3]. For example, a physical object passing by an RFID reader too fast, such that the reading can not be completed results in not recording the moving object.

Further, the view maintained of the workflow system is not representing the actual flow of the physical goods. E.g. in [4] the authors argue that information in a workflow system can be imprecise. Furthermore, the workflow in the physical world may deviate from the workflow in the workflow system. Discrepancies can be either exceptional, i.e., ad hoc changes in the real world on request which are not reflected in the workflow system, or structural, i.e., the deviation/evolution of the realized workflow execution from the workflow specification [5].

In this paper, a traditional way of using sensor data for decision making in workflows is compared to the proposed way of using sensor data to identify inconsistencies between physical and digital world. Further, the implications of imprecisions on the sensor and/or the workflow side in

Figure 1. Logistics Scenario: Picking and Sorting

handling physical goods are discussed. Correlating sensor data for decision making in a workflow means using sensor information at decision points of a workflow (see Sect VI) - further called **tight correlation**. Correlating sensor data for identifying inconsistencies with a workflow means relating sensor data to states of a workflow execution (see Sect VII) - further called **loose correlation**. It turns out that besides classical syntactic and semantic integration problems [6], several challenges can be identified (see Sect IX): In case of tight correlation the benefit is that the usage of sensor data reduces the human involvement in the workflow execution, while it increases the vulnerability to incomplete modeling of physical objects, e.g., changes in the sensor infrastructure or applying the workflow to additional physical objects. In case of loose correlation, the benefit is a high flexibility to errors and changes in the infrastructure and the evolution of the workflow, while the involvement of humans in executing the workflow increases.

II. SCENARIO

As a running example, part of a logistics process is used. It consists of a conveyor belt, equipped with sensors for weight measuring, height and length scanning of packages, and a bar code reader on the packages (see Fig 1). The conveyor belt connects a container used for shipping and a warehouse. The belt is used in two different workflows: the loading of a container called *picking* and the unloading of a container called *sorting*. The sensor data acquired on the belt correspond to the physical movement of boxes in the physical world, which corresponds to processing information of a purchase or procurement order related to the loading or unloading of the container and the corresponding changes in stock values in the warehouse information system.

In a realistic scenario, a company has several of these conveyor belts operated in parallel for several containers either operated for sorting or picking (see lower part Fig 2 representing the physical world). Workflow models use

the picking and sorting as building blocks (tasks) for order and procurement workflows. Workflows are described as workflow schemas which are instantiated as potentially several concurrent workflow instances (see upper left part of Fig 2)[7]. The state of a workflow instance describes the view of the workflow instance on a subset of the physical world.

Sensor data also distinguish schema and instances [8]. A schema is the specification of relations by specifying columns using names and data types, while instances contain the actual data values. Sensor data here consist of rather static metadata and streaming sensor data (see upper right part of Fig 2). Sensor data instances describe the view of a sensor on a subset of the physical world at a specific point in time.

The movement of boxes in the context of sorting and picking is a task performed in the physical world, which can be observed by the sensors installed on the conveyor belt (see Fig 1: dimension scanner, scale and bar code reader) and which has an associated representation in a workflow instance. In the following three scenarios illustrate potential incorrect data in sensor data or workflow systems.

- 1) Sensors can break or require maintenance. As a consequence, the weight sensor can be out of order, but due to heavy working load the conveyor belt must be used anyway. This results in weight sensor data of value Not a Number (NaN).
- 2) The packages are unloaded by human workers. The worker places two packages with the same bar code on top of each other on the belt. As a consequence, the length and weight sensors are not able to detect that there are two packages on top of each other. The bar code reader reports only a single code. The two packages are represented in the sensor data as one package with double weight and wrong height.
- 3) The vendor of the packages in the container calls a human operator telling him that five packages are not included in the container, but will be delivered the next day by overnight express. The operator calls the supervisor in the warehouse informing him about the late delivery of the five packages. The related procurement workflow instance remains unchanged since changing it is much more effort than keeping track of this late delivery outside the system. Thus, the workflow system does not know about the late delivery of the five packages.
- 4) The workflow evolves by applying the workflow to additional products, i.e., physical objects. In case these objects do not have a bar code but an RFID chip, the lack of an RFID reader and a bar code results in incomplete sensor data.

In the following notions of states are introduced for workflow systems (Sect III) and sensor data (Sect IV),

which are then correlated with each other (Sect VI and VII). The above three scenarios are used to investigate workflow properties for incorrect data.

III. NOTIONS OF WORKFLOW STATES

A workflow is specified as a workflow schema and executed as a workflow instance [7]. The instance follows the specification provided by the schema and maintains a notion of state. Execution of a workflow instance is based on a sequence of state changes, also called state transitions. Depending on the underlying workflow model state is represented very differently.

One extreme type of workflow models maintains all state information as global variables, like e.g. Tuple spaces [9], [10] or I/O automata [11]. In this case a state transition means changing global variables. Whether a specific state transition is allowed in a particular state is expressed as constraints on the global variables.

The other extreme type of workflow models maintains all state information implicitly in state names. Here a state name does not have any meaning, but can only be reached by a specific sequence of state transitions like e.g. Finite State Automata [12] and Petri Nets [13]. Thus, the actual meaning of the state is hidden in state transition sequences resulting in the particular state.

BPEL as a workflow specification in the SOA context provides the expressiveness for pure global variable and pure state name workflow models, as well as for mixed workflow models.

In general, state name workflow models can be transformed into global variable workflow models, by introducing a history global variable maintaining the list of executed state transitions as a representation of state. Thus, in the following, a **workflow state is represented as a set of variables** representing the state information.

State transitions change a workflow from one state to another. Transitions can be enabled based on external or internal events, which are represented as transition labels. External events are e.g. incoming messages or events initiated by a user or a sensor, while internal events are e.g. timeouts or data driven decisions.

With regard to the scenario in Sect II there are two schemas depicted in Fig 2 one representing the picking and one the sorting. Further there are two instances of the sorting and three instances of the picking in different states, where grey circles mark the states executed so far.

A. Infrastructure Support

In the following it is discussed how current workflow systems provide access to events and state information. Inclusion of external events, like e.g. sensor data is straight forward. In case of BPEL an event could be issued as a message sent to the BPEL process (push) requiring correlation

Figure 2. Underlying Model: physical world, workflows and sensor data

information, or it could be requested from the sensor in case a decision point has been reached (pull).

Accessing state information is less easy. In the general reference model for workflow systems [14] accessing state information is not standardized although this has been considered beneficial for distributed environments. This is to a certain extend understandable, since the representation of state information is essential for the performance of the workflow system. However, vendor proprietary, potentially closed, interfaces exist for sharing state information between several distributed workflow systems.

This observation concur with the findings in service related workflow systems, like an attempt on monitoring a BPEL workflow engine [15] or on monitoring approaches in service compositions [16], [17], [18].

There are options to access state information if workflow systems based on tuple spaces are used. In [19] an XML based tuple space is described which can be used to explicate state information and make it accessible via standard read and write operations. An example of a distributed workflow engine facilitating tuple spaces is described in [10]. In general workflow systems store state information in a database (e.g. [20]), which can be made accessible via services like e.g. [21]. This seems to be applicable especially in enterprise scenarios, like the one outlined in Sect II, since it is a closed enterprise system and accessing state information without modifying it allows to relate workflow instance state and physical world.

IV. NOTION OF SENSOR DATA STATE

Sensor data are observations by a sensor of a physical object. Often raw sensor data requires further processing to higher level events to be useful in the workflow context. Sensor data fusion combines metadata and other sensor data to events on a higher abstraction level or of higher data quality [22]. The sensor data from the scenario (see Sect II) are fused into tuples describing physical packages with length, height, weight, and code information. In the fusion process, the continuous height signal is used to derive the actual height of the package and the length assuming a fixed speed of the conveyor belt. The maximum of the continuous weight signal is derived as the weight of the package. The bar code signal needs no modification. Due to the time proximity of sensor readings and the knowledge on the location of the sensors and the direction in which the conveyor belt is turning, the individual sensor data can be grouped into tuples.

Relations between objects in the physical world and sensor data require a possibility to relate physical objects and sensor data by means of an identifier. In the example, the bar code is a non unique identifier of a package in the physical world. The fusion of sensor data allows to associate length, height, and weight with the identifier and therefore with a physical world object. Since these codes are not unique, several packages with the same Universal Product Code (UPC) may be contained in the container and the warehouse. Thus the association of sensor data and physical objects is not unique.

Although sensor data fusion is an important topic, the focus in this paper is on relating sensor data and workflow systems. Therefore, the difficulties of sensor data fusion are not addressed. Since fused sensor data has the same characteristics as acquired sensor data, it is assumed that sensor data in sufficient data quality and sufficient abstraction is available. Further, since metadata do not reflect the dynamics of the physical world, the focus in the following is on streaming sensor data.

The state of the physical world as observed by the available sensors at a point in time is described by the combination of the readings of all available sensors at that point in time. The dynamics of the physical world corresponds to changes of data over time. Thus, a state change in sensor data is the change on the sensor state between two points in time.

With regard to the scenario in Sect II the metadata are represented on the upper part of the right side, while the actual sensor data is depicted on the lower part of the right side. Data and metadata of height sensor, the scale and the bar code reader are depicted for a time t_0 plus the specified delay.

A. Infrastructure Support

The Sensor Web Enablement is a working group in the Open Geospatial Consortium. In this context the *Observations and Measurements*¹ specification describes schema for sensor data and acquisition methods, which are used by the *Sensor Observation Service*² specification providing a Web Services interface for retrieving sensor data. Besides these standards there are many proprietary Web Service and REST interfaces for devices, like e.g. [23], [24]. In general there are two main concepts on how sensor data can be made accessible as a Web Service based stream.

The first approach hides the physical sensor devices by using wrappers and providing a virtual sensor with web based data publishing and data access capabilities. Further, but less standardized and potentially more distributed approaches are the Global Sensor Network [25], the Sensor Data Web³, and the dataturbine⁴ [26]. In the context of the Open Grid Forum, the Web Service Data Access and Integration [24] specification has to be mentioned, which describes web based interfaces to data resources. In the context of the *Internet of Things* activities different wrapper based infrastructures have been proposed to make sensor data available as Web Services, like e.g. [27].

The second approach is based on Web Service enabled sensor devices, like e.g. the sunspot⁵ using a Java based Web Service stack. A survey of Web Service toolkits based on

several different languages and a comparative performance test can be found in [28]. Also in context of the Internet of Things and Smart Factories web based sensor data access has been discussed either based on Web Services [29] or on REST interface [23].

V. CORRELATION OF SENSOR DATA AND WORKFLOW SYSTEMS

A conclusion of the previous two sections is that accessibility of state information of sensor data and workflow systems is available - at least in an enterprise context. The sensor state and the workflow state each represent a subset of the state of physical objects in the physical world at a point in time. The aim is now to integrate and correlate the overlapping parts of sensor data state and workflow state via the state of physical objects.

A. Assumptions

This paper is not focusing on syntactic and semantic integration problems [6], but addresses a systematic investigation on how sensor data and workflow state information can be correlated.

Since the correlation can only be accomplished via physical objects these objects must be distinguishable, i.e., identifiable. In particular, the physical objects have to be identifiable on the same level of granularity. For example, the modeling of a pallet full of packages in a workflow system may not know details about the packages on the pallet. If the sensor data represent the packages and do not know about the pallet, then the granularity differences can not be resolved.

Thus, it is assumed that sensor data and workflow system have no syntactic and semantic integration problems, and being based on identifiable physical objects on the same granularity level, the correlation of states is possible.

B. Correlation

Sensors can detect the presence or properties of a physical object, however, not every change in location or property is detected by sensors due to unreliability of sensors, maintenance, or other obstructions. The presence of a physical object can be related to a physical location using sensor metadata (see Fig 3 center).

State transitions in a workflow instance result either in a change of location or property of a physical object or no change at all. However, not every change on a physical object is directly related to a state transition (see Fig 3 right hand side). For instance if for some cleaning activity of a shelf in the warehouse the packages have to be moved from the shelf, this is not related to any order or procurement state transition.

This means that the focus must be on observable changes of physical objects. Further, a change in location or property of a physical object is not always sensed and not always

¹<http://www.opengeospatial.org/standards/om>

²<http://www.opengeospatial.org/standards/sos>

³<http://sourceforge.net/projects/sensordataweb/>

⁴<http://www.dataturbine.org/>

⁵<http://www.sunspotworld.com/>

Figure 3. Sensor Data and Workflow Correlation

related to a state transition, which makes correlating sensor data and workflow systems a challenge (see Fig 3 left hand side).

In general, there are two possibilities to relate sensor data and workflow systems:

- **State transitions** in workflows can be triggered by an event, which can represent a state change in the physical world observed by sensors. Thus, sensor data is used in the control flow of the workflow as a *trigger* of a state transition.
- Changes of **workflow states** and sensor data states can be correlated to see whether changes on physical objects are consistently performed over all physical objects effected by a state transition and are consistently observed by the sensor data. This is a *monitoring* approach to identify inconsistencies in sensor data and workflow states.

Error classes:

- successful workflow execution in case of reliable sensor data without exception: this is the case where everything goes fine
- successful workflow execution in case of unreliable/erroneous sensor data: e.g. see Sect II error 2
- successful workflow execution in case of workflow processing with undocumented changes in the workflow: e.g. see Sect II error 3
- effect of workflow evolution: e.g. see Sect II error 4
- effect of changes on the sensor infrastructure: e.g. see Sect II error 1

C. Evaluation Criteria

The investigation of the two correlation mechanisms is done according to the following criteria:

- manual labor cost during workflow schema design: How much manual effort is initially required to integrating the sensor and workflow systems
- manual labor cost during workflow instance execution: How much manual effort is required for the execution of a single workflow instance?

VI. SENSOR STATE AND WORKFLOW STATE TRANSITIONS

A. Approach

A workflow state transition is triggered by an external event, i.e., a change in a sensor reading. In particular, an

event is defined by a predicate P evaluated on workflow and sensor state information. The state information is included since the predicate may contain workflow instance specific variables, like e.g. a package number for package tracking. Checking whether the predicate evaluates to true or false can be done either on the workflow side (pull of sensor data) or on the sensor data side (push of sensor data). There are approaches considering sensor data quality assessments provided with the actual sensor measurements. This is the most commonly used way of correlating sensor data and workflow systems.

B. Assessment

The approach is assessed according to the criteria in Sect V-C. The event is defined as a predicate over the workflow instance state and the sensor data state. This means that the workflow designer needs to have a good overview of existing sensor data and a good understanding of their semantics as well as metadata. Defining the predicate and ensuring that it is unique for the targeted workflow instance requires high manual effort. The advantage is that after the predicate has been defined the state transition is automatically triggered and therefore no manual intervention during workflow execution is required.

The execution of the event triggered workflow state transition strongly depends on the reliability of the sensor data and the strict execution of the workflows. If data is unreliable or erroneous the state transition may not be triggered potentially causing the workflow instance not to terminate. If there is changes in the workflow, which effects the handling of physical goods potentially used for triggering a state transition, then this again may lead to non terminating instances.

In general the tight connection of sensor data and workflow instances increases the risk of deadlocked workflow instances in case of changes on the workflow schema or the available sensor data (this includes physical deployment of sensors as well as processing of derived sensor data). The risk of deadlocks can be reduced by specifying exception handling potentially involving human operators. Changes in a workflow may effect the path of physical objects and the quality and availability of sensor data at a specific location may influence the evaluation of the event predicates. Since only a subset of the sensor data is used in an event predicate and only some of the triggers are based on sensor data the risk is assessed medium.

A summary of the assessment for the different criteria discussed before is contained in *State Transition* column of Table I.

VII. SENSOR STATE AND PROCESS STATES

A. Approach

The state of physical objects as represented in sensor data and workflow system are correlated. The movement

Criteria	State Transitions	Workflow State
manual labor cost during workflow schema design	high	none
manual labor cost during workflow instance execution	none	maybe
execution with reliable data without exception	good	good
execution in case of unreliable/erroneous data	medium	good
execution in case of workflows with changes	medium	good
effect of workflow evolution	medium	none
effect of changes on the sensor infrastructure	medium	none

Table I
SUMMARY OF THE ASSESSMENT

of physical objects captured in sensor data and the state changes in workflow systems are not dependent on each other, thus comparing states indicates whether states are consistent or not. Therefore, workflow state changes are used as synchronization points for checking consistency, because they occur less often than sensor data state changes.

The sensor data state changes in the time span between the previous workflow state change and the current workflow state change are considered to correlate sensor data and workflow instance. Let's assume a workflow instance has reached state s at time n_1 and a state transition is performed based on state s resulting in state s' at time n_2 . Then the change of state as a consequence of the transition can be expressed as a set of insert, delete, and update operations on the description of the state s . The sensor state is the set of all sensor readings between time n_1 and n_2 . The state changes are related by a mapping function representing the correlation of changes in the workflow state and the sensor data state.

Since the focus of this paper is on investigating correlation mechanisms, in the following only an intuition on the mapping function is provided. A detailed investigation is left for future work. The change in a workflow state could result in three mapping functions each representing a specific moving pattern of physical objects effecting sensor data.

- moving: all physical objects of a workflow activity are moved in the same way and to the same location in the physical world, i.e., all objects are recorded by the same sensors.
- sorting: all physical objects arriving in a container are sorted and distributed in a warehouse. In the workflow instance state this means assigning location IDs to each physical object, while in the physical world the objects are treated independently and thus observed by different sensors.
- picking: physical objects are moved from the warehouse in a container for shipment. The physical objects of various locations are moved into a single location, while this means an update operation on the warehouse stock and an update on the packing list of the container,

the movement of the objects is visible in the sensor state.

B. Assessment

The approach is assessed according to the criteria in Sect V-C. The sensor data and workflow system are not directly connected, therefore there is no manual effort required during design time for combing them. Since the workflow system is not directly correlated with sensor data, state transitions can not be enabled by sensor data. Thus, other forms of coordination are required instead, which may include coordination by humans resulting in low manual labor. Since workflow system and sensor data are not directly connected there is no affect of sensor data on the workflow execution. However, the mapping is effected by unreliable data and by workflows with changes. An inconsistent mapping indicates that workflow state and sensor data are not describing the same state of the physical world, which might indeed be the case. Inconsistent mappings require attention from an administrator, but do not stop the workflow execution. Ad hoc changes on the workflow schema or the sensor infrastructure result in a different path respectively different detection of physical objects, which changes the mapping function characteristics but does not effect the workflow execution.

A summary of the assessment for the different criteria discussed before is contained in *Workflow State* column of Table I.

VIII. RELATED WORK

In [30] the authors propose the definition of subprocesses which are selected dependent on the state of context information. A selection of a subprocess corresponds to a state change triggered by context information. Specific subprocesses for exception handling dependent on context information are proposed in [31].

In [1] the authors extend BPEL to incorporate sensor data facilitating sensor push, pull and choice operations. The approach is extended in [3] considering systematic and stochastic sensor data errors by allowing data quality measures in the event predicate definition. The data quality information has to be provided by the sensor data infrastructure.

In [32] physical objects are associated with tasks in the context of a specific workflow, like e.g. to borrow a book from a library. These tasks correspond to subprocesses changing the global workflow state. A more user centered view on services issued by objects is discussed in [33], where the physical presence of objects is translated in available services available to the user on his mobile device. The detection of a physical object results in a state change of the user interface on the user's mobile device to offer these additional services.

Tasks can also be outsourced to physical objects with computational capabilities called devices as e.g. discussed in [34]. This approach is extended in [35] to service compositions which are provisioned by an infrastructure depending on the discovered services provided by devices. In this case, subprocesses are outsourced, which results in sharing service descriptions rather than triggering or sharing state information as discussed in this paper.

Up to my knowledge, the idea of loosely coupling sensor data and workflow states has not been discussed in literature before. There is however, quite some literature on correlating different models based on their overlap describing a system. Examples are correlating business process models and coordination models at design time [36] or at run time [37], or the monitoring of Web Service compositions using log information [18]. However, sensor data and workflow systems are indirectly correlated via the state of the physical world, while the afore mentioned approaches do the correlation directly.

The mapping function between sensor data and workflow state has some similarities with data integration approaches like e.g. [6] for homogeneous models. In particular, the approach of probabilistic data integration could be a good source of inspiration [38].

Please be aware that although dealing with sensor data scientific workflows are very different from business workflows as discussed in this paper. Scientific workflows are focusing on processing data often coordinated by data [39], [40].

IX. DISCUSSION

A conclusion from Tab I is that the correlation of sensor data to workflow states is much more robust with regard to change, unreliable data and workflow changes. However, this comes with the price of potentially requiring other events for state transitions, which might require human interaction.

For both approaches it has to be considered that a sensor reading of a physical object might belong to several workflow instances. An example are inter-organizational workflows, where several workflows are coordinated by exchanging messages, potentially resulting in exchange of physical objects being relevant for several workflow schema and at least one instance per schema.

Another challenge for both approaches are non uniquely identifiable physical objects. For example several physical boxes with the same product bar code are handled by a picking and a sorting workflow instance at the same time. In case of triggering a transition with non unique identifiable objects the definition of the predicate gets more difficult since the correlation information to the workflow instance has to be included. In case of correlating states the mapping of sensor data states and workflow instance states gets more complicated, since state changes of physical objects are hard to identify because a set of readings belongs to a set of indistinguishable objects.

A challenge for state correlation is the synchronization of workflow state changes with sensor state changes. Since there are many workflow instances running in parallel and physical objects potentially being relevant for several workflow instances, it is difficult to identify the relevant sensor state space and its changes between two synchronization points of a workflow instance.

X. CONCLUSION

In this paper, a tight and a loosely correlation of sensor data and workflow systems is investigated. It turns out that the tight coupling, where sensor data is used for workflow coordination, may reduce manual effort for coordinating workflow instances. However, the workflow execution is much more dependent on a stable infrastructure and the reliability of sensor data and workflow systems. The loose coupling has not been investigated in this context although similarities to data integration have been identified.

Interesting challenges are the mentioned synchronization problems of workflow state changes and sensor data changes and the handling of non unique identifiable physical objects in parallel running workflow instances each using a subset of these objects.

REFERENCES

- [1] M. Wieland, O. Kopp, D. Nicklas, and F. Leymann, "Towards context-aware workflows," in *CAiSE07 WS Proc Vol.2*, 2007.
- [2] R. Jedermann, C. Behrens, D. Westphal, and W. Lang, "Applying autonomous sensor systems in logistics—combining sensor networks, rfids and software agents," *Sensors and Actuators A: Physical*, pp. 370 – 375, 2006.
- [3] M. Wieland, U.-P. Käppeler, P. Levi, F. Leymann, and D. Nicklas, "Towards integration of uncertain sensor data into context-aware workflows," in *GI Jahrestagung*, 2009, pp. 2029–2040.
- [4] P. Soffer, "Mirror, mirror on the wall, can i count on you at all? exploring data inaccuracy in business processes," in *Proceedings of BPMDS*, 2010.
- [5] R. Lenz and M. Reichert, "It support for healthcare processes - premises, challenges, perspectives," *Data Knowl. Eng.*, vol. 61, no. 1, pp. 39–58, 2007.
- [6] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *VLDB J*, vol. 10, no. 4, pp. 334–350, 2001.
- [7] W. Aalst and K. Hee, *Workflow Management - Models, Methods, and Systems*. MIT Press, 2002.
- [8] M. Kifer, A. Bernstein, and P. Lewis, *Database Systems - An application-Oriented Approach*, 2nd ed. Pearson Education, 2005.
- [9] D. Gelernter, "Generative communication in linda," *ACM Transactions on Programming Languages und Systems*, vol. 7, no. 1, pp. 80–112, 1985.

- [10] J. Yu and R. Buyya, "A novel architecture for realizing grid workflow using tuple spaces," *Intl WS on Grid Computing*, pp. 119–128, 2004.
- [11] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1996.
- [12] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2001.
- [13] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, 1981.
- [14] D. Hollingsworth, "Workflow management coalition - the workflow reference model," WfMC-TC-1003, 1995.
- [15] A. Slomiski, "On using bpel extensibility to implement ogis and wsrfl grid workflows: Research articles," *Concurr. Comput. : Pract. Exper.*, vol. 18, no. 10, pp. 1229–1241, 2006.
- [16] A. Keller and H. Ludwig, "The WSLA framework: Specifying and monitoring service level agreements for web services," *J. Network Syst. Manage.*, vol. 11, no. 1, 2003.
- [17] H. B. Newman, I. Legrand, P. Galvez, R. Voicu, and C. Cirstoiu, "MonaLisa : A distributed monitoring service architecture," *CoRR*, vol. cs.DC/0306096, 2003.
- [18] L. Bodenstag, A. Wombacher, M. Reichert, and M. C. Jaeger, "Monitoring dependencies for SLAs: The mode4SLA approach," in *IEEE SCC*, 2008, pp. 21–29.
- [19] R. Tolksdorf and D. Glaubitz, "Coordinating Web-based systems with documents in XMLSpaces," in *CoopIS*.
- [20] M. Reichert and P. Dadam, "Enabling adaptive process-aware information systems with adept2," in *Handbook of Research on Business Process Modeling*. Hershey, New York, 2009, pp. 173 – 203.
- [21] A. Wombacher, "Composable data processing in environmental science - a process view," in *Proc Intl Conf on Services Computing*, 2008, pp. 499–502.
- [22] D. L. Hall and J. Linas, *Handbook of Multisensor Data Fusion*. CRC Press, May 2001.
- [23] D. Guinard and V. Trifa, "Towards the web of things: Web mashups for embedded devices," in *Second Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web*, Madrid, Spain, Apr. 2009.
- [24] M. Antonioletti, A. Krause, N. W. Paton, A. Eisenberg, S. Laws, S. Malaika, J. Melton, and D. Pearson, "The ws-dai family of specifications for web service data access and integration," *SIGMOD Rec.*, vol. 35, no. 1, pp. 48–55, 2006.
- [25] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," *Mobile Data Management, 2007 International Conference on*, pp. 198–205, May 2007.
- [26] T. Fountain, S. Tilak, P. Hubbard, P. Shin, and L. Freudinger, "The open source dataturbine initiative: Streaming data middleware for environmental observing systems," 2009.
- [27] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. M. S. de Souza, and V. Trifa, "SOA-based integration of the internet of things in enterprise services," in *ICWS*. IEEE, 2009, pp. 968–975.
- [28] G. Moritz, S. Prueter, D. Timmermann, and F. Glatowski, "Deployment of embedded web services in real-time systems," *Scalable Computing: Practice and Experience*, vol. 10, no. 3, pp. 265–275, 2009.
- [29] L. M. S. de Souza, P. Spiess, D. Guinard, M. Köhler, S. Karnouskos, and D. Savio, "SOCRADES: A web service based shop floor integration infrastructure," in *Intl Conf Internet of Things*, 2008, pp. 50–67.
- [30] S. Modafferi, B. Benatallah, F. Casati, and B. Pernici, "A methodology for designing and managing context-aware workflows," in *Mobile Information Systems II*. Springer, 2005, pp. 91 – 106.
- [31] M. Adams, A. Hofstede, W. Aalst, and D. Edmond, "Dynamic, extensible and context-aware exception handling for workflows," in *Proc CoopIS*, 2007, pp. 95–112.
- [32] P. Giner, C. Cetina, J. Fons, and V. Pelechano, "Developing mobile workflow support in the internet of things," *IEEE Pervasive Computing*, vol. 9, no. 2, pp. 18–26, 2010.
- [33] G. Broll, E. Rukzio, M. Paolucci, M. Wagner, A. Schmidt, and H. Hussmann, "Perci: Pervasive service interaction with the internet of things," *IEEE Internet Computing*, vol. 13, no. 6, pp. 74–81, 2009.
- [34] S. Karnouskos, D. Guinard, D. Savio, P. Spiess, O. Baecker, V. Trifa, and L. M. S. de Souza, "Towards the real-time enterprise: Service-based integration of heterogeneous soa-ready industrial devices with enterprise applications," in *Proc Symp on Information Control Problems in Manufacturing*, 2009.
- [35] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Transactions on Services Computing*, Feb. 2010.
- [36] Z. Zlatev and A. Wombacher, "Consistency between e³-value models and activity diagrams in a multi-perspective development method," in *OTM Conferences (1)*, 2005, pp. 520–538.
- [37] L. Bodenstag, A. Wombacher, M. U. Reichert, and R. Wieringa, "Monitoring Collaboration from a Value Perspective," in *Intl Conf on Digital Ecosystems and Technologies*, 2007.
- [38] M. van Keulen, A. de Keijzer, and W. Alink, "A probabilistic XML approach to data integration," in *ICDE*. IEEE Computer Society, 2005, pp. 459–470.
- [39] W. Tan, P. Missier, R. K. Madduri, and I. T. Foster, "Building scientific workflow with taverna and BPEL: A comparative study in cagrid," in *ICSOC Workshops*, 2008, pp. 118–129.
- [40] B. Ludäscher, M. Weske, T. M. McPhillips, and S. Bowers, "Scientific workflows: Business as usual?" in *BPM*. Springer, 2009, pp. 31–47.