Annotating Business Processes with Usage Controls

Andreas Schaad
SAP Research, Security & Trust
Vincenz-Priessnitz Str. 1
76131 Karlsruhe
andreas.schaad@sap.com

Anja Monakva
SAP Research, Security & Trust
Vincenz-Priessnitz Str. 1
76131 Karlsruhe
ganna.monakova@sap.com

ABSTRACT

Complex Supply Chain interactions provide an ideal example of interconnected physical and logical assets that require protection. More specifically, we observe an increasing demand for specifying and enforcing usage control policies within supply chains, relating to both physical + as well as logical assets.

In this paper we will highlight some possible usage control scenarios. We will present our existing control visualization framework to position the identified usage controls in the more general context of safety and security controls. We provide an in-depth discussion of the key constructs of our model and how they can be used to specify and visualize usage controls.

Keywords

Usage Control, Workflow, Security, Visualization

1 Introduction

Consider a typical supply chain between a supermarket, a producer of deep-frozen goods and a logistics provider. Possible usage control scenarios may be to:

- observe a certain temperature during shipment
- not store the shipment next to household cleaning agents
- allow authorized changes of a shipment / purchase order after release
- delete shipment data after completion of shipment
- retain and handle audit relevant shipment data appropriately

Those five scenarios already indicate that we need certain contextual information for specification and later enforcement of usage controls. Equally, we observe that we quite naturally spoke about "the shipment", sometimes referring to a potential physical asset such as a palette, sometimes referring to a logical business object such as a purchase order.

This requires us to consider a conceptual model that would be capable to provide the needed context to define appropriate usage controls as well as an associated execution semantics that can provide support for usage control enforcement.

In this paper we discuss our existing control visualization framework [1] that allows specifying security and safety controls over logical and physical assets. We will then discuss this model in the context of usage controls with a focus on their visualization. A set of possible usage controls will then be analyzed, together with possible mechanisms supporting their specification and enforcement.

2 Control Visualization Framework

Our Control Visualization Framework (CVF) consists of a supply chain risk database; an extended workflow specification language; as well as a workflow execution engine. Our framework explicitly addresses the visualization of safety and security controls on the workflow model and as such we now address the possible integration of usage controls.

2.1 Basic Language Constructs

Figure 1 shows the concepts and their relations used in our supply chain language. A supply chain model is represented by a choreography that contains multiple internal processes represented through activities (hierarchical activities). A choreography specification can contain a number of variables which are basically the representation of the supply chain assets. Variables can be annotated with tags, which identify certain properties of the assets. Each process can have a number of In/Out/InOut arguments, whereby each argument will refer to the variable and therefore to an asset used in the choreography. Output and Input arguments can be connected with a Connector, which specifies the transition of the corresponding asset from one process to another.

2.2 Usage Control Extensions

Overall, our discussion will address how enterprise context can be used for specification of usage control policies at "design-time", as well as how enterprise context can be used at "run-time" to make appropriate usage control decisions. On basis of our simple, yet powerful control visualization model, we now consider its extension with respect to usage controls. We base our discussion along the three core dimensions of usage control [2], namely addressing the data provider and data consumer; provisions and obligations controls; as well as obligation enforcement through signalling and monitoring.

Variables and Tags

Variables essentially describe the assets in our supply chain, and we distinguish between logical assets (such as a purchase order or customer file) and physical assets (such as a physical palette of goods). Tags are then assigned to variables and classify an asset, for example, a purchase order, as audit relevant or the actual shipped good requiring careful handling. This implies that providers will assign the tags to the asset and consumers have to act accordingly when receiving the asset. We, however, now do not only distinguish between data providers and consumers, but rather between asset providers and consumers.

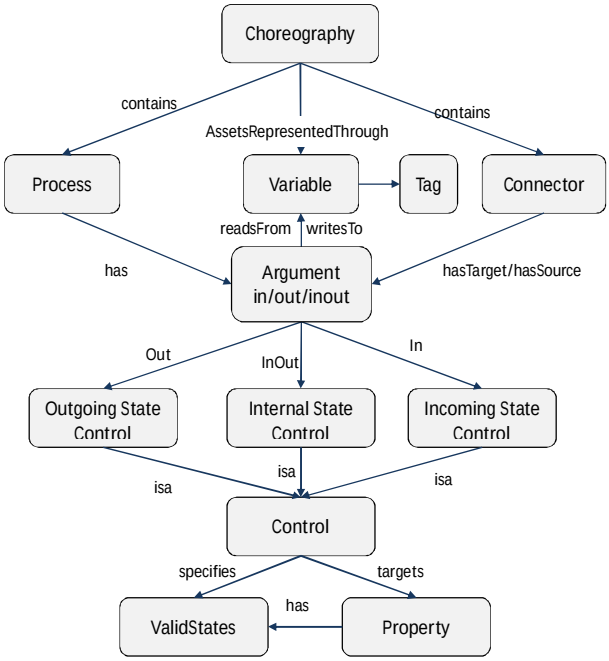

Figure 1: Concepts used in the SCM language

This distinction does not have an impact on provision and obligation controls. We rather observe that in a supply chain context, there are multiple stages where either an asset provider articulates an obligation and where that obligation turns into a provision in a subsequent step of the supply chain.

In other words, the initial step in the supply chain is not necessarily the point in time where all usage controls are defined; this may happen throughout the supply chain. The later enforcement of obligations is driven by the tags and any monitoring needs to be done in accordance with the properties defined for a tag (i.e. frozen goods must be consistently stored at -17 degrees).

Controls and process steps

Controls generically bundle a set of properties, for example, a digital signature control will provide integrity and non-repudiation while a temperature control will provide just temperature. What is important with respect to asset usage is that controls can be enforced at three different states of a process step – input, output, and internal.

This distinction proves to be highly relevant for usage controls, as they relate to the point in time where an asset consumer will accept and then enforce the obligations articulated by the asset provider. For example, if the provider of a logical asset defines that the data in a purchase order must be handled in a confidential manner, then the consumer of this data will check at the input state of the "receive order" process step whether he is able to do this and if so, he will need to enforce this obligation during the internal state of the following "process order" step. Equally, the data consumer will need to restate the obligation at each output state of a process step.

Different control points exist depending on the type of argument. In-arguments can only have input controls that can check the state of the asset before the activity (process) starts its execution; Out-arguments can only have output controls, which check asset state after an activity completed its execution; InOut-arguments can have input, output, as well as internal controls, which control the state of the asset during the activity execution.

2.3 Usage Control Specification Approach

We will now describe how usage controls can be defined together with other safety and security controls in a supply chain. Figure 2 shows the conceptual model of our presented approach. The three main concepts in the model are Asset, Threat and Control. An asset has potential threats and certain controls can countermeasure these threats. The role of the rest of the model is to help identify which threats are applicable to which type of asset and which controls can be used to countermeasure these threats. In the following we describe the steps of our control specification approach based on this conceptual model, specifically focusing on usage controls.

Asset identification

In this step we identify the assets used in a business process that requires controlled execution. As discussed earlier, we identify two types of assets: "Logical assets" representing critical business data such as purchase order details or credit card numbers, while "physical assets" represent real world objects used in the business process, such as a shipment in the supply chain. Any asset can be described by a set of "Properties" it possesses. For example, a logical asset can be described by a set of properties such as signature or encryption properties. Similar, any physical asset can be described by a set of properties such as temperature, location or size. Each property is defined by a set of "states" it can adopt. For example, a temperature property can be in a state −18° C or +5° C, while a signature property can be in the state Unsigned, SignedNoModification, etc.

An asset is characterized by the set of properties it has and the state(s) each property has at a current point in time. This combination between asset type as well as property and state appears to support usage controls. Different types of usage controls can be defined for either a logical or physical asset, but more importantly, we can define what expected properties an asset must exhibit over its lifetime, directly supporting later monitoring and enforcement of usage controls.

Asset classification

It is not sufficient to only distinguish between logical and physical assets, but each asset must be classified. Different threats are applicable to different assets depending on an asset classification. For example, two logical assets can have different threats: the first logical asset might contain private information about a customer with a threat of information disclosure, while another logical asset might contain financial data, which has threat of unauthorized modification. Similar, a frozen physical asset might have threat of being stored at an excessive temperature, while a fragile physical asset may be subject to a threat of being broken. To allow a business process designer to classify business assets, the concept of a "Tag" has been introduced. A tag attached to an asset identifies a certain characteristic or classification of this asset. Figure 4 shows an example set of tags that can be used to classify logical and physical assets. Tags can be attached to the assets in a business process, which would promote awareness of the asset characteristics used in the process.

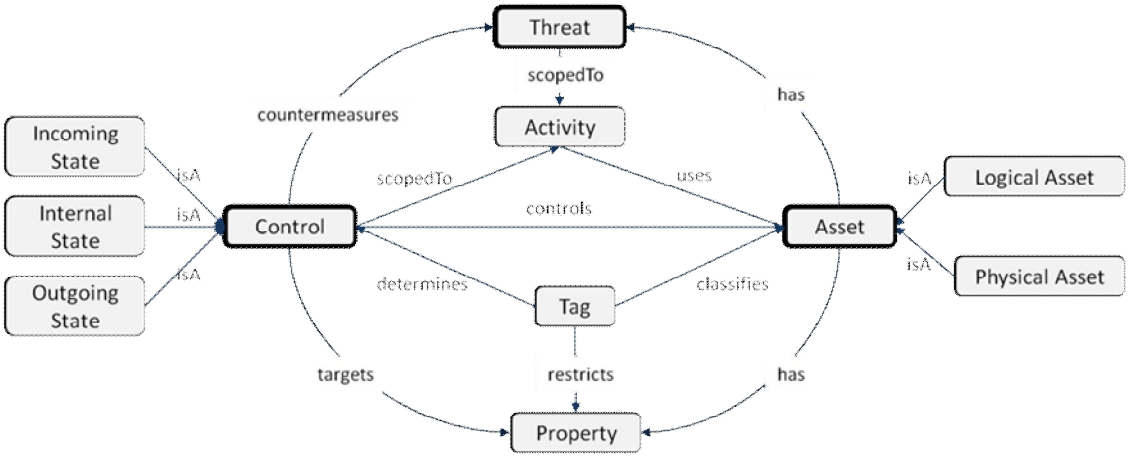

Figure 2: Conceptual Model

In a supply chain example, an Ice Cream asset can be annotated with the tags DeepFrozen and LightSensitive, while a PurchaseOrder can be annotated with the tags AuditRelevant and Financial.

This tagging or classification of assets would have an immediate effect on the definition of later usage controls. For example, if we tag a purchase order as audit relevant, then this would imply a later control over the retention period. Another example could be a customer record asset, tagged as personal information, which would in turn require privacy-aware handling throughout the supply chain process.

Controls identification

To provide a generic methodology for relating controls to the assets, we classify controls based on the asset properties they can control. For example, a temperature property can be monitored by a temperature sensor control. Similar, a signature property can be controlled by a signature service that can identify whether the document is signed and whether the signature is valid (monitor), or sign the document (enforcer). We distinguish between state-properties and range-properties, and correspondingly state-controls and range-controls. State properties are specified by a list of states a property can take and a state control contains specification of valid states for this property for a given asset at certain time. Range properties are defined by the range of the values it can take, and a range control contains the border specifications for the property values that a certain asset can have at a certain time. Each tag attached to an asset can be viewed as a restriction on certain asset properties. A tag puts restrictions on a property by restricting the set of valid states for this property and the related asset. For example a DeepFrozen tag puts constraints on the temperature property of a physical asset by restricting valid temperature values to under −18° C. Based on tags and implied property restrictions, controls can be identified. For example for the DeepFrozen tag a temperature control will be suggested. To achieve consistency in control identification, the required controls are identified based on the rules stored in a database. The rules derive required controls for each activity that uses an asset annotated with certain tags. Thereby controls can depend on multiple tags as well as on the type of activity that uses the asset. For example, an asset that is flammable and explosive might require different controls than only flammable assets. Controls are implementations of a certain functionality that can control a certain property. A temperature sensor can control a temperature property, while a service that can sign and validate digital signatures is able to control signature property. Figure 3 gives an overview over sample controls for logical and physical assets.

This now again emphasis how our model and reference implementation could handle usage controls at policy specification as well as later runtime. Based on a certain tag (such as audit relevant) and asset type (purchase order) we automatically derive the appropriate usage controls such as guaranteed retention time by deletion only after 10 years.

As mentioned above, controls are related to a certain asset property rather than to an asset, which allows to use the same controls with different assets that have the same property. A signature or encryption service can be used with multiple logical assets, as well as sensors can be used with multiple physical assets. A control “understands” a certain property and can be configured with the valid states for the property and the given asset. The role of the control is to ensure that the asset property the control is responsible for is in a valid state. For example, for a deep-frozen pizza we need controls to ensure that the pizza temperature is under −18° C. Controls are scoped to activities, therefore different activities that use the same assets can have different controls applied to the same assets. Controls can be divided into three main categories – Monitors, Enforcers and Auditors:

- Monitors observe the state of a certain asset property in a specified activity. It can notify a violation in case an invalid state has been detected, display the current states in a dashboard and log them into a database.
- Enforcers are used to transform the state of a property. For example, a signature property enforcer can automatically sign created documents, while a temperature property enforcer might be able to switch on an emergency freezer if the temperature monitor detects that the current temperature is too high.

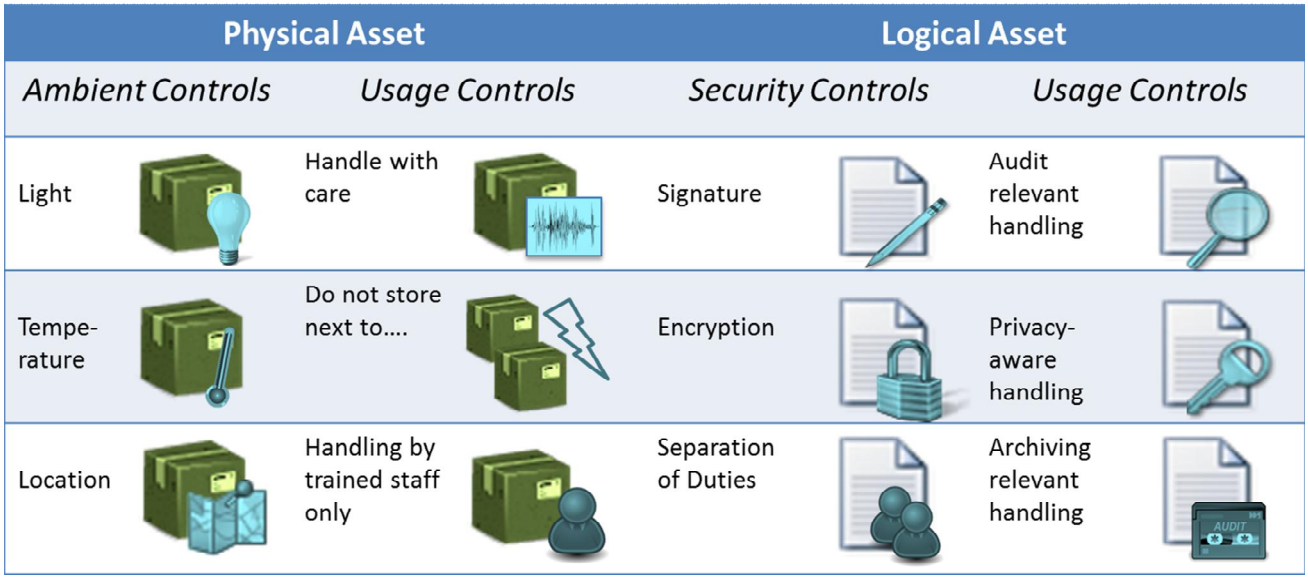

Figure 3: Example Controls & Visualizations

- Auditors generate reports on property state history. Auditors use data logged in by the monitors to compute specified functions. For example, an auditor can analyze if the temperature of an asset was above limit for longer than 5 minutes and correlate this information with the asset location.

Control points identification

A control can be applied at different stages of an activity execution. If applied on activity initialization, it can control the incoming states of the asset properties; if applied on activity execution, it can control the internal states of the asset properties; if applied on activity completion, it can control outgoing states of the asset properties. Depending on the type of activity, different control types are applicable. Incoming state controls and outgoing state controls can be enforced by the workflow engine – it can invoke control services to verify that the asset properties are in a correct states and can for example suspend a workflow (or execute any other activities that are defined as part of a reactive process) if a violation has been detected. The internal controls on the other side can be viewed as the requirements on the activity implementation with regard to the asset handling. Having such requirements as part of the model can be used for example for generation of contracts between participants from the designed process model. The next section describes how the presented approach has been realized in a prototype.

3 Architecture

Figure 4 gives an overview over the architecture in a SOA environment. At the design time, the RiskDB is consulted to identify threats and countermeasures for the business process assets that have been classified with the tags. At the runtime, process execution engine invokes control services at the specified control points through the control service broker. All controls are available as property control services that subscribe to the property they can control in the RiskDB, specifying the type of the control (monitor, enforcer, or assessor) and the assets it can handle. A business process engine sends the asset or asset reference and the property to control to the control service broker, which then looks up available services in the RiskDB and finds a service that can evaluate or change the state of the given property for the given asset. For example, a sensor platform will find the sensor that is attached to the given asset, and a signature service suitable for the given document type will be selected. All property states, as well as process execution states are stored in a LogDB, which feeds data into the dashboard and allows offline analysis of the completed instances and improvement of the rules specified in RiskDB.

4 Implementation

Our current prototype is based on Windows Workflow Foundation (WF 4.0). Figure 4 shows the prototype architecture, where the bold elements represent our extensions to the WF4.0 framework.

Microsoft Workflow Foundation uses variables to represent data used in a business process, however, the variables are defined in a variable tab and are not visible in the designer. To advocate security awareness, we extended existing workflow modelling constructs with two visual elements for logical and physical assets. Furthermore, we added an asset (or variable) panel to the business process, which contains all assets used in the process. To add a new asset (variable) to the process, the user just needs to drag & drop the corresponding visual element into the asset/variable panel of the workflow. To enable asset classification we provided a tag toolbar. To annotate a variable the user needs to drag & drop the corresponding tag from the toolbar onto the visual asset specification now present in the asset panel. By combining different tags, a user can specify different characteristics of an asset. Figure 5 shows a screenshot of the ice cream supply chain process modelled using our tool. It contains two variables that can be seen in the right panel: An IceCream variable annotated with a DeepFrozen and LighSensitive tags and a PurchaseOrder variable annotated with Financial and AuditRelevant tags. On basis of these tags we would now define the possibly required usage controls.

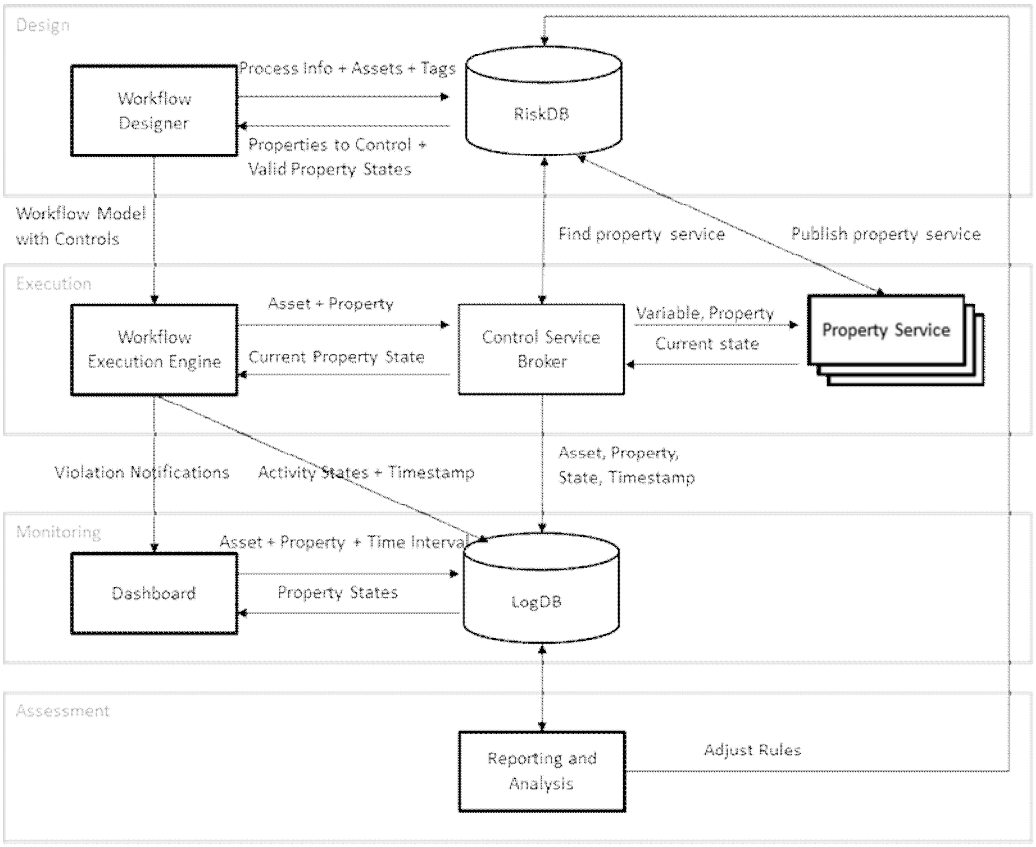

Figure 4: Architecture

In Figure 5 we can see four activities: Order, Dispatch, Transport, and Receive activities. The Activity Order outputs PurchaseOrder, which is then passed as an input argument to the Dispatch activity. The Dispatch activity then outputs IceCream, which is passed to Transport activity and then through the Transport activity to the Receive activity. Depending on the type of argument (In, Out or InOut), we can see different types of control points available for each asset in each activity. This allows the user to define input state controls on the incoming asset states (PurchaseOrder in Dispatch activity) output controls on outgoing asset states (PurchaseOrder in Order activity), and internal controls on data that exists all the way through activity execution (IceCream in Transport activity). These would be the points in the supply chain execution where usage controls would be enforced and monitored.

To identify controls required to countermeasure potential threats or usage control requirements, we developed a Risk Database (RiskDB). The RiskDB stores relations between asset tags, threats these tags imply for different activities, and controls that should be applied to such assets in each activity. When a user annotates an asset with a new tag, a query is sent to the RiskDB that selects the necessary protection measurements (or controls) for each activity that uses this asset. After this the tool checks if the controls are already present in the model and if not, shows an error with the information about missing controls. This requires a business designer to model secure processes with respect to the rules stored in the RiskDB.

To enable control specification, we provide a control toolbar. To identify at which point of activity execution a control must be applied, the user needs to drop a control into the corresponding container. In Figure 5 we can see an output signature control applied to the PurchaseOrder variable in Order activity. This control specifies that the data must be signed when it leaves this activity. In the Dispatch activity we can see an example incoming state control that states that the PurchaseOrder signature property must be in state verified to be used by this activity. In the Transport activity internal temperature and light controls are specified, which define that IceCream temperature must be between −50°C and −25°C and light must be under 200 Lumen. Additional controls could be added as input and output controls.

In general, any number of controls can be applied to each asset in each activity. For example, a possible usage control on the Purchase Order asset could be that the supermarket chain ordering the ice cream asks the icecream manufacturer to not share any non-relevant details of the order (eg price) with the logistics provider. This would then imply that at execution time, the purchase order file is sanitized, ie the usage control would be placed on the purchase order asset at the outgoing asset state.

Another example could be a usage control demanding that the logistics provider deletes all shipment data after 60 days. In this case, we would place a control on the outgoing purchase order asset state in the Transport activity which would eventually trigger a timed deletion event.

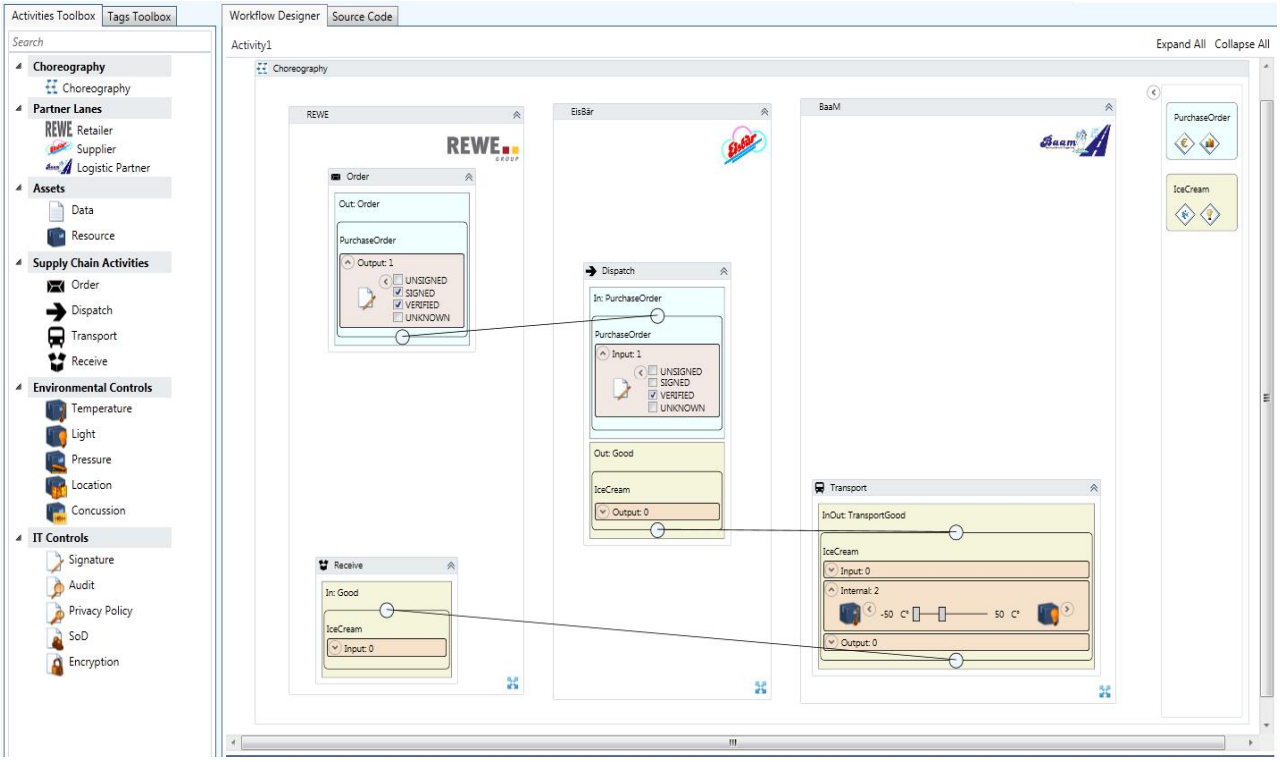

Figure 5: Screenshot of a Modelled Supply Chain & Example of Applied Controls

5 SUMMARY AND CONCLUSION

In this paper we discussed our existing control specification framework [2] and its possible extension in the context of usage controls. One key finding was that we can equally specify usage controls on physical as well as logical assets. Tagging assets in our supply chain does allow automatically inferring appropriate controls and then enforcing them at workflow execution time. We demonstrated how we envision the later visualization of usage controls.

Of course many points remain open and require further research, though we consider them to be outside the scope of this paper. For example, while certain controls are quite straight forward to automatically implement (eg. a simple digital signature) other controls appear to require more contextual information, both at specification and runtime, and we need to consider a possible application of our earlier transformation approaches [3].

Another point is further required work on more fine-grained usage control taxonomies as there appears to be no existing work on basis of which we could provide a more detailed visualization of controls. The presented visualizations in this paper are of course rudimentary and would require involvement of the HCI community such as seen in earlier SOUPS workshops. We however envision that next generation UI framework such as HTML 5 or MS WPF will allow definition of more interactive (usage control) policy Widgets. For example, we could consider widgets that actually incorporate selection boxes, pull-down menus or input fields.

Future work will now look into associating specific usage control mechanisms to our business process and control visualization platform. The PrimeLife policy engine [4] should allow us to specify and then enforce privacy-specific usage control policies. Sanitizable signature schemes [5] could support allowed modification of signatures depending on intended usage and supply chain state. Provable data possession schemes [6] could be used to articulate usage control requirements such as “only process order if you have obtained a safety clearance”.

REFERENCES

[1] Ganna Monakova, Achim D. Brucker and Andreas Schaad. Security and Safety of Assets in Business Processes. In ACM Symposium on Applied Computing (SAC), ACM Press, 2012

[2] Hilty, M., Pretschner, A., Basin, D., Schaefer, C., Walter, T.: A Policy Language for Usage Control. 12th European Symp. on Research in Computer Security (ESORICS), pp. 531-546, Dresden, September 2007

[3] Christian Wolter, Michael Menzel, Andreas Schaad, Philip Miseldine, Christoph Meinel: Model-driven business process security requirement specification. Journal of Systems Architecture - Embedded Systems Design 55(4): 211-223 (2009)

[4] Slim Trabelsi, Jakub Sendor, Stefanie Reinicke: PPL: PrimeLife Privacy Policy Engine. POLICY 2011:184-185

[5] G. Ateniese, D. Chou, B. de Medeiros, and G. Tsudik. Sanitizable signatures. In ESORICS’05, 2005.

[6] Giuseppe Ateniese, Randal C. Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary N. J. Peterson, Dawn Xiaodong Song: Provable data possession at untrusted stores. ACM Conference on Computer and Communications Security 2007: 598-60