A Simulation Model for Evaluating Distributed Storage Services for Smart Product Systems

Markus Miche[1], Kai Baumann[1], Jerome Golenzer[2], and Marc Brogle[1]

[1] SAP Research Switzerland,
Kreuzplatz 20, 8008 Zurich, Switzerland
{markus.miche,kai.baumann,marc.brogle}@sap.com
[2] EADS Innovation Works,
Rue Marius Terce 18, 31300 Toulouse, France
jerome.golenzer@eads.net

Abstract. While subsets of the functionality of intelligent physical objects such as context awareness or integration with backend systems can be analyzed and assessed using small-scale experiments, the evaluation of most distributed services such as content replication and placement strategies requires dedicated simulation environments. Despite the availability of various related simulation frameworks for analyzing P2P overlay networks, WSNs, or RFID technology, there is no simulation model that reflects the characteristics of *smart* products with embedded computing, storage, and networking functionality. This paper proposes a simulation model that facilitates simulation of distributed storage services of smart products based on P2P overlay networks. In order to exemplify the suitability of the proposed simulation model, it is applied to the industry application scenario *smart aircraft manufacturing* as envisaged by EADS Innovation Works.

Keywords: Smart Products, Simulation, Distributed Storage, P2P, Aircraft Manufacturing.

1 Introduction

In diverse domains such as manufacturing, retail, or logistics, one can encounter an increasing number of intelligent physical objects that enhance business processes by means of real-time data or context-aware and personalized user guidance. This ranges from objects being equipped with smart labels such as RFID or NFC tags to so-called *smart* products with embedded computing, storage, and networking capabilities. Smart products are able to build ad-hoc networks as well as to self-organize in scalable Peer-to-Peer (P2P) overlay networks that facilitate efficient communication without relying on central entities. Moreover, smart products make use of distributed process models in order to assist and interact with their users as well as to autonomously collaborate to fulfill their tasks. For this purpose, smart products operate complex distributed services such as distributed process execution services or distributed storage services including content replication and placement strategies.

A. Puiatti et al. (Eds.): MobiQuitous 2011, LNICST 104, pp. 162–173, 2012.

However, even though smart products represent high-class intelligent physical objects, they typically only possess limited on-board storage and computing resources. Moreover, due to their mobility and usage in different environments with varying environmental conditions, smart products are subject to regular disconnections. This characteristic is further affected by the usage of energy-efficient communication modules, which automatically deactivate themselves after a certain period of inactivity [1].

While functionality such as context awareness or integration with backend systems can be analyzed and assessed by means of small-scale experiments, the evaluation of complex distributed services requires extensive test environments. Especially in early development phases, the installation of large test beds is not appropriate and simulation environments reflecting characteristics of smart products appear to be the approach of choice. Simulation environments can be classified into *network simulators* and *overlay simulators*. Network simulators such as NS-3[1] or OMNeT++[2] facilitate packet-level simulation of network protocols and can be utilized for analyzing wireless sensor networks or RFID systems. Overlay simulators such as OverSim[3] or PlanetSim[4] abstract from network details and focus on overlay network routing protocols and services (see survey presented in [2]). However, to the knowledge of the authors, while both kinds of simulators could be used as basis for simulating and evaluating distributed storage services of smart products, there is no simulation model that fully reflects the characteristics of smart product systems.

This paper presents an extended overlay simulation model that facilitates simulation of distributed storage services of smart products being organized in P2P overlay networks. It reflects the characteristics of smart products including their heterogeneity, resource limitation, mobility, as well as their process-based operations. As opposed to common overlay simulators that assume simplified topologies, the proposed model enables consideration of underlay networks with realistic topologies in order to enable accurate simulations. Even further, the paper presents a detailed analysis and modeling of the smart products industry scenario *smart aircraft manufacturing* as envisaged by EADS Innovation Works using the proposed simulation model. This shows not only the suitability of the latter but moreover a potential procedure for analyzing and modeling simulations of distributed storage services of smart products.

The remainder of the paper is structured as follows: Section 2 presents the proposed simulation model for smart products. This includes the conceptual structure of the model, simulation events, product and content properties, as well as underlay network configuration options. Thereafter, the smart products industry scenario *smart aircraft manufacturing* as well as a detailed modeling of the latter using the proposed simulation model are presented in Section 3. The paper concludes in Section 4 with an outlook on future work.

---

[1] http://www.nsnam.org/
[2] http://www.omnetpp.org/
[3] http://www.oversim.org/
[4] http://projects-deim.urv.cat/trac/planetsim/

2 Conceptual Simulation Model

2.1 Conceptual Structure of the Simulation Model

The proposed extended overlay simulation model for distributed storage services of smart products consists of four logical layers. The underlay network layer is used to model basic underlay networks with realistic topologies in order to enable accurate simulations and analyses of distributed services. This includes node distribution and clustering, bandwidth allocation, as well as communication delays. Due to the purpose of the simulation model, i.e., the simulation of distributed storage services of smart products being organized in P2P overlay networks, additional underlay network properties such as packet delay variation or packet loss are not considered. This way, quality attributes of distributed storage services such as average content access latencies or content access hit rates given that messages are assigned pre-defined timeouts can be evaluated with much higher accuracy compared to simulation models that purely rely on virtual random topologies. The second layer, overlay network, encapsulates all functionality of P2P content location and routing substrates.

While most simulation models solely consider a single application layer, the proposed model distinguishes between overlay services, i.e., distributed services that directly operate on the P2P overlay network, and the actual application logic that utilizes functionality of the overlay service layer. Finally, the simulation model applies the Common API (CAPI) as defined by [3] in order to ease simulation of different overlay networks and overlay services. The layered structure of the simulation model as well as an exemplary instantiation is presented in Fig. 1.

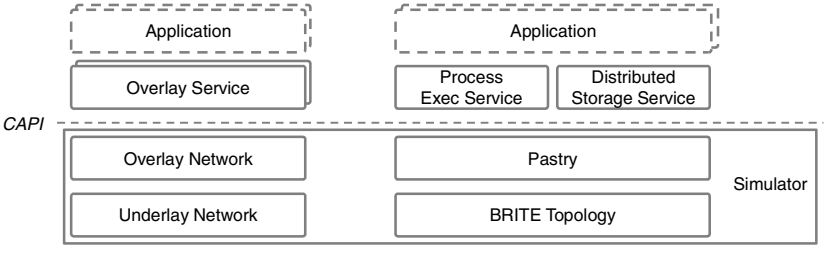

Fig. 1. Conceptual Structure of the Simulation Model

The simulation model is designed for discrete event simulators and follows the two phases and the control flow defined by [4]. While events scheduled and processed by the simulation engine affect all layers of the simulation model (see Section 2.2), only the two lower layers are actually controlled by the simulation engine and require simulator-specific implementations. In contrast, applications as well as overlay services that comply with the CAPI can directly be deployed in the simulation environment without requiring adaptations. This simplifies simulation of distributed services of smart products and is especially valuable, if simulation is used in early development phases and complemented by test-bed experiments.

2.2 Simulation Events

The dynamics of smart product systems is modeled by events that are scheduled and processed by the simulator. On the underlay and overlay network layer, the simulation model supports node *join*, node *fail* and *leave*, as well as node *move* events. Joining nodes are integrated into the underlay network, in which they are associated with unassigned underlay nodes that reflect their capabilities. Moreover, they self-organize into the overlay structure according to the means of the concrete overlay network implementation. Node *fail* events are used to model ungraceful leaves of nodes, i.e., nodes that suddenly disappear (e.g., because they run out of energy or loose connectivity). This behavior is complemented by node *leave* events, which are used to model graceful leaves of nodes that explicitly announce their leave to enable other nodes to react accordingly (e.g., by triggering content handover). Finally, node *move* events trigger node position changes, which are fully covered by the underlay network layer (see Section 2.3).

In addition to the typical events for simulating node churn, the simulation model covers overlay service events for simulating distributed storage services. This includes content *get* as well as *put* events, which reflect queries for content objects and explicit content storage, respectively. Replication or caching strategies are not covered by *put* events and must be separately estimated in order to determine the expected overall load generated during simulation runs. As described in [5], the simulation model moreover includes *process execution* events that trigger process operations and – indirectly – distributed storage service events for collecting content required during process executing as well as for persisting process results (an example is presented in Section 3.1).

The distribution functions of node *join* and *move* events as well as content *get* and *put* events are configured as described in [6, 7]. Node *leave* and *fail* events adopt the lifetime churn model defined by the simulation framework OverSim.[5] Finally, node *process execution* events are scheduled based on a normal distribution (see Table 1).

Table 1. Simulation Events

Event	Distribution Function
Node join	$t_{join} \coloneqq Poisson(\lambda)$
Node fail	$t_{fail} \coloneqq Weibull(\alpha, \beta)$ $P(fail) = \delta$
Node leave	$t_{leave} \coloneqq Weibull(\alpha, \beta)$ $P(leave) = 1 - P(fail) = 1 - \delta$
Node move	$t_{move} \coloneqq Normal(\mu, \sigma^2)$
Get request	$t_{get} \coloneqq Poisson(\lambda)$
Put request	$t_{put} \coloneqq Poisson(\lambda)$
Node process execution	$t_{process} \coloneqq Normal(\mu, \sigma^2)$

Both overlay network and overlay service events are scheduled according to a node-centric event scheduling approach. Hence, instead of having the simulator randomly assigning events to nodes, nodes are responsible for scheduling their events. As depicted in the simulation event graph illustrated in Fig. 2, node *join* events are

[5] http://www.oversim.org/wiki/OverSimChurn

scheduled during simulation initialization in order to prepare a basis network structure. *Join* events schedule node-specific *move*, *put*, *get*, and *process execution* events as well as themselves to enable new nodes joining the network. Each of these events reschedules itself in order to simulate node-specific activities according to the distribution functions defined above. Finally, since nodes may either fail or leave the network, node *fail* and *leave* events are assigned probabilities that are evaluated during the scheduling phase of the node join event routine.

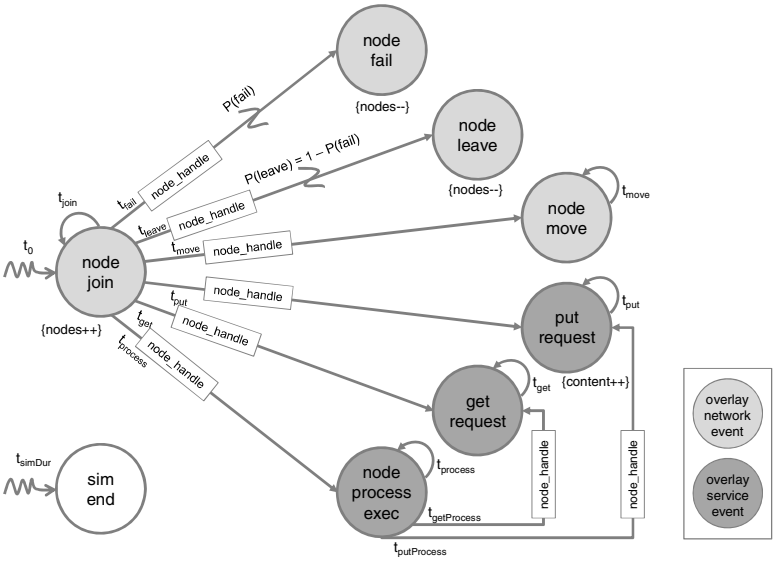

Fig. 2. Node-Centric Simulation Event Graph

2.3 Simulation Parameter

In addition to simulation events, the proposed simulation model includes node- and content-related simulation parameters that reflect the specifics of smart products. This enables modeling of heterogeneous node classes, which differ in the configuration of the distribution functions presented in Table 1 as well as regarding their on-board storage capacity. Heterogeneous communication capabilities are modeled by associating node classes with matching underlay nodes, i.e., nodes with few resources are assigned to underlay nodes with low-bandwidth communication links. Moreover, multiple content classes can be modeled and assigned to node classes, with each content class consisting of content objects of different size. This is required, because smart products with limited resources typically generate and request content of relatively smaller size than powerful smart products.

In order to facilitate simulations with high numbers of nodes and content objects, both storage capacity and content size are defined in virtual storage units (e.g., Integer values). Finally, for each content class, access popularity is configured using a Zipf-like distribution function [7].

Table 2. Simulation Parameter

Simulation Parameter	Distribution Function
Node storage capacity	$Finite-range\ discrete\ distribution$
Content size	$Finite-range\ discrete\ distribution$
Content access popularity	$Zipf-like\ distribution\ (expontent < 1)$
Underlay hierarchy	$2-layer\ hierarchy$
Underlay cluster positioning	$Randomly$
Underlay intra-cluster node positioning	$Heavy\ tailed$
Underlay cluster topology	$Scale-free\ network$
Underlay intra-cluster topology	$Star\ topology$
Underlay node bandwidth	$Uniform\ distribution$
Underlay link delay	$Assigned\ by\ topology\ generator$
Underlay node movement	$Intra-\ and\ inter-cluster\ movements$

The underlay network used within the proposed simulation model requires at least as much underlay nodes as the maximum amount of nodes that may join the network during simulation (see Table 1). Nodes are organized in multiple autonomous systems (AS), which are used to model node clustering. This reflects the typical spatial clustering of smart products known from most envisaged application scenarios. Each AS consists of at least one AS-router that enables intra- as well as inter-AS communication and is assigned a configurable upper bound of participating nodes. To enable inter-AS communication given node churn and movement, inter-AS topology is realized as a scale-free network using the topology generation model introduced by Barabási and Albert [8]. Underlay nodes within an AS are arranged in a star topology, with the AS-router representing the central node. While AS are randomly placed on a virtual map, intra-AS distribution of underlay nodes follows a heavy-tailed distribution. Hence, inter-AS communication passes at least two AS-routers, namely the local AS and the remote AS-router. Communication between two underlay nodes of the same AS is always routed via the local AS-routers. This is reasonable, because the simulation model solely considers smart products being organized in P2P overlay networks; ad-hoc connections between products is out of scope.

Delivery time of route messages sent in the underlay network is affected by link delays and the smallest bandwidth available on the path between sender and receiver. While the former mainly affects delivery time of small messages (e.g., *get* request), the latter has an impact on messages with large content objects (e.g., *get* response). Since the main purpose of the underlay network layer is the realization of an accurate simulation of overlay networks and services, it only makes use of a static delay and bandwidth matrix, which is created before the actual simulation. This way, the delay and the minimal available bandwidth of a path between any two nodes are statically available and do not need to be calculated lazily. While this reduces simulation accuracy, it clearly enhances simulation execution.

Finally, node movement is realized by absolute changes of nodes' underlay position (as opposed to considering node traces). This is reasonable, since distributed storage service likely don't consider any small location change but focus on complete movements after which nodes remain stable for a certain period of time (e.g., for triggering content replacement). The simulation model supports local and remote

position changes to enable intra-AS and inter-AS node movement, respectively. Note that instead of keeping idle underlay nodes for node *move*, *fail*, and *leave* events to enable proper communication failure handling, the simulation model maintains an internal mapping between overlay nodes and underlay nodes (both idle and active). This reduces the overall number of underlay nodes required to reflect a certain scenario with mobile products.

3 Smart Aircraft Manufacturing

3.1 Application Scenario Description

Today's aircraft manufacturing processes are characterized by paper-based process descriptions that have to be carried by blue-collar workers as well as manual tracking of assembly results. Obviously, the integration of smart products such as smart torque wrenches has great potential for enhancing efficiency of aircraft manufacturing processes. This includes automation of process steps such as tool configuration, maintenance of results of single process steps, as well as accumulation and aggregation of assembly results in order to obtain (real-time) information about aircrafts' manufacturing status.

Fig. 3 provides an overview of a future *smart aircraft manufacturing* scenario as envisaged by EADS Innovation Works [9]. The illustration shows the different kinds of smart products used in the scenario as well as their organizational interrelation.

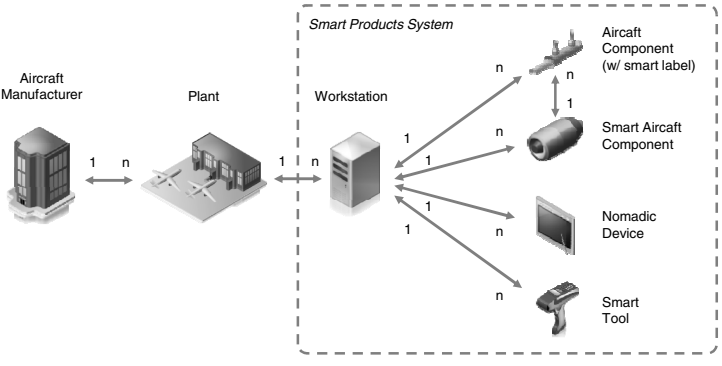

Fig. 3. Smart Aircraft Manufacturing

As described in [10], instance-level aircraft manufacturing process descriptions are distributed from the aircraft manufacturer to the different plants, in which they are further dispatched to the corresponding smart aircraft components and/or workstations at which manufacturing is accomplished. Plants consist of several workstations with multiple blue-collar workers per workstation. Each blue-collar worker carries a nomadic device, which is used to display instructions of manufacturing processes as well as to annotate corresponding results. In addition, each blue-collar worker makes use of smart tools (e.g., smart torque wrench, smart drill) that are automatically

configured based on extended manufacturing process descriptions and capable of delivering results (e.g., torque and angle) to nomadic devices. Even further, while most components such as bolts are merely identifiable via smart labels, certain (compound) aircraft components also play the role of smart products. This way, they are able to maintain digital representations of their manufacturing status as well as open manufacturing processes that are to be accomplished (see [11]). Eventually, the component-level information maintained by smart aircraft components and workstations is delivered to and aggregated by the aircraft manufacturer in order to realize digital representations of entire aircrafts as being manufactured [9].

From a technical perspective, the smart products of the smart aircraft manufacturing scenario, i.e., nomadic device, smart tool, smart aircraft component, as well as workstations, are interconnected in a structured, multi-level P2P overlay network. Due to the limited storage capacity of smart products and the incomplete network coverage in plants [10], a distributed storage service is employed as overlay service with dedicated content replication and placement strategies. This services aims at maintaining a high-level of user-perceived performance as well as at enhancing content availability and durability in order to ensure complete digital representations of aircrafts as manufactured that can be used in subsequent phases of the product lifecycle [5].

3.2 Simulation Modeling

In order to enable evaluation of different overlay networks and services as well as the overall benefit of the envisaged smart aircraft manufacturing scenario, the latter has been modeled using the proposed simulation model.[6] The model simulates 3 shifts à 8 hours in a single plant.[7] This plant is assumed to consist of 10 workstations with a maximum number of 100 blue-collar workers per workstation. Each worker is assumed to carry 1 nomadic device as well as up to 3 smart tools. To simplify simulation modeling, workstations are assumed to be used to assemble one smart aircraft component at the same time (sequential production). Non-smart aircraft components are not explicitly modeled as they are assumed to not being capable of participating in the overlay network (e.g., due to resource and processing limitations). Instead, they use other smart products as proxies for submitting *put*/*get* requests. These requests are indirectly considered by adapting the request rate of the associated smart aircraft components. Finally, it is assumed that the assembly of smart aircraft components lasts 2 hours on average; the storage unit used to model storage capacity of smart products as well as content size represents 0,1 MB.

In order to determine the scale of one simulation step, multiple simulation runs with an average content size of 10 storage units (1 MB) and different underlay network configurations (e.g., different network size and node positioning) were performed and analyzed. This resulted in an average roundtrip time of 20 simulation steps for a *get*

[6] Since there is no prototype of the presented scenario available so far, the configuration of most parameters is based on assumptions that reflect the current status of work.

[7] Limiting the scope to a single plant is valid, because the focus of the simulation is on the extent to which smart products enhance performance of aircraft manufacturing processes.

request. Based on the assumption of an average delivery time of 10 seconds for a message with a payload of 1 MB, a simulation step is scaled to 0,5 seconds. Consequently, the simulation of 3 shifts à 8 hours lasts 172.800 simulation steps.

Simulation Events and Parameter. First, workstations are modeled as stable nodes that join during simulation initialization phase; they neither fail nor leave the network, nor do they change position. Workstations possess an average storage capacity of **127.500** MB. Second, based on the assumption of an average number of **90** blue-collar workers per workstation with each worker carrying **1** nomadic device, the simulation model consists of **900** nomadic devices on average. Nomadic devices possess on-board storage with an average storage capacity of **5.400** MB and have an average lifetime (in terms of network participation) of **4** hours, which reflects a single break per shift. Third, due to the assumption of sequential production, the simulation model includes **10** aircraft components on average (one per workstation). As described above, aircraft components have an average lifetime of **2** hours and an average storage capacity of **32,5** MB. Finally, based on the assumption of each blue-collar work carrying **2** smart tools on average, there is an average number of **1.800** smart tools with an average storage capacity of **1,6** MB. Due to their limited resources, smart tools have an average lifetime of **35** minutes. This reflects the fact that smart tools tend to deactivate communication in order to save energy in case they are not actively used (e.g., after completion of a process they have been used for).

Since all activities in the above-described smart aircraft manufacturing scenario are based on well-defined process models, the determination of *put*/*get* request rates per node class is derived from the process execution configuration. It is assumed that processes consist of 8 steps with each step requiring the usage of 2 smart tools (average values). As presented in Table 4, the average execution time of processes is defined as 30 minutes. Processes are solely started by nomadic devices and each nomadic device is only capable of starting a single process at the same time. Moreover, it is assumed that each blue-collar worker performs 3 processes during the assembly of an aircraft component. While smart tools store results of each process step, nomadic devices, workstations, as well as aircraft components only store results of completed processes.

Consequently, because smart tools have an average lifetime of roughly one process duration, they submit 8 *get* requests (step-related configuration) and 8 *put* requests (storage of step result) on average. Nomadic devices are alive for approximately the assembly of 2 aircraft components. Since they store both processes as well as corresponding results they have an average *put*/*get* request rate of 12. Due to an average number of 90 blue-collar workers per workstation with each worker performing 3 processes to assembly an aircraft component, the latter has an average *put* request rate of 270 per lifetime. Moreover, since it is assumed that aircraft components know not only their manufacturing status but also open processes to be accomplished, their average *get* request rate equals their average *put* request rate. Finally, workstations have an average *put*/*get* request rate of 3.240. On the one hand, they provide nomadic devices and/or aircraft components with processes to be accomplished. On the other hand, they accumulate and aggregate results of all local

processes in order to obtain digital representations of aircrafts as being manufactured. The detailed modeling of content class and node classes is presented in Table 3 and Table 4, respectively.

Table 3. Modeling of Content Classes in Storage Units (* indicates assumptions)

	Values (Portion)	Average Content Size
Content Class A*	2 (0,4), 5 (0,3), 10 (0,2), 50 (0,1)	9,3
Content Class B*	2 (0,8), 5 (0,2)	2,6

Table 4. Modeling of Node Classes (* indicates assumptions)

	Workstation	Nomadic Device	Aircraft Component	Smart Tool
#Nodes (average)*	10	900	10	1.800
Storage capacity*	750.000 (0,3) 1.500.000 (0,7)	20.000 (0,2) 50.000 (0,6) 100.000 (0,2)	250 (0,7) 500 (0,3)	10 (0,7) 20 (0,2) 50 (0,1)
Node join*	*Join during simulation initialization*	$Poisson(28.800)$	$Poisson(14.400)$	$Poisson(3.600)$
Node fail*	*Stable*	$Weibull(15; 28.800)$ $P(fail) = 0{,}02$	$Weibull(10; 14.400)$ $P(fail) = 0{,}05$	$Weibull(4; 4.200)$ $P(fail) = 0{,}1$
Node leave*	*Stable*	$Weibull(15; 28.800)$ $P(leave) = 0{,}98$	$Weibull(10; 14.400)$ $P(leave) = 0{,}95$	$Weibull(4; 4.200)$ $P(leave) = 0{,}9$
Node move*	*Stable*	$Normal(7.200; 100)$	*Stable*	$Normal(3.600; 100)$
Put request[8]	$Poisson(53{,}3)$ *content class A*	$Poisson(2.384)$ *content class A*	$Poisson(52{,}6)$ *content class B*	$Poisson(425)$ *content class B*
Get request	$Poisson(53{,}3)$ *content class A*	$Poisson(2.384)$ *content class A*	$Poisson(52{,}6)$ *content class B*	$Poisson(425)$ *content class B*
Node process execution*	–	$Normal(3.600; 100)$	–	–

Underlay Configuration. To determine the required total amount of underlay nodes, it has to be taken into account both the maximum number of products and a buffer to perform node movement. As mentioned before, the network of a single plant consists of $\mathbf{2.720}$ smart products on average with $\mathbf{20}$ products (workstation, aircraft component) being non-moving. Due to the variance of the selected distribution functions (see Table 1), the maximum number of nodes participating in the network is $\mathbf{5.420}$. To ensure that intra- and inter-AS node movement can also be performed during peak phases, the total amount of underlay nodes is incremented to $\mathbf{6.000}$.

[8] In order to ensure overlay stabilization up to a certain degree before submitting *put*/*get* requests, the latter are not submitted but after an initial delay of 200 simulation steps. This value results from analyses of multiple simulation runs and has to be adapted depending on overlay network properties.

According to the number of workstations, the underlay network is organized in 10 ASs. Each AS has a maximum number of 600 nodes. This includes up to 599 underlay nodes with one of them playing the role of a workstation and at least one AS-router. The bandwidth of inter-AS and intra-AS communication links is configured using a constant bandwidth of 1.000 MBits/s and a uniform distribution between 11MBits/ and 54MBits/s, respectively.

The smart aircraft manufacturing scenario has been modeled using the topology generator BRITE [12] with the parameters shown in Table 5 and Table 6. While most attributes can be directly realized using BRITE, the star topology used to interconnect underlay nodes with local AS-routers cannot be modeled by default. This has been approached by a modified delay and bandwidth matrix, which includes links between each underlay node and its local AS-router. Links that are not part of the generated topology are added using the minimum bandwidth and the sum of all delays along the shortest path between the underlay node and its local AS-router.

Table 5. BRITE Underlay Configuration

	BRITE Underlay Configuration
Topology type	$Top-Down, 2\ layer\ hierarchy$
AS connection model	$Smallest\ k-degree\ (k=25)$
Topology generation model	$Barabási, Albert\ (incremental\ growth, preferential\ connectivity)$

Table 6. BRITE Underlay Node Configuration

	AS Layer	Node layer
Size main plane	1.000	100
Size inner plane	100	10
Number of nodes	10	600
Node placement	$Random$	$Heavy\ tailed$
Min number of links per node	9	2
Bandwidth distribution	$Constant$ (1.000)	$Uniform\ (min=11, max=54)$

4 Conclusion and Outlook

This paper proposes a discrete, event-based simulation model that enables accurate simulation and evaluation of distributed storage services of smart products. For this purpose, it extends common overlay simulators and includes underlay network configurations in order to enable modeling of realistic topologies. Moreover, by complying with the Common API, the model eases the simulation of different P2P content location and routing substrates as well as overlay services. Being tailored to the simulation of distributed storage services, the simulation model provides overlay network and overlay service events, which are scheduled according to a node-centric scheduling approach. Moreover, in order to account for the characteristics of smart products, the simulation model contains simulation parameter that enable modeling of heterogeneous node and content classes as well as proper underlay network configurations. Finally, the paper presents an application of the proposed simulation

model to the smart products industry scenario *smart aircraft manufacturing* as envisaged by EADS Innovation Works. This includes a description of the scenario as well as detailed modeling of simulation events and parameters, and illustrates the suitability of the simulation model.

Based on the simulation model and the presented instantiation, which are currently both being implemented, future work focuses on simulation and evaluation of distributed storage services and P2P content location and routing substrates of smart products. Amongst others, this includes the replication strategy presented in [5] as well as the hybrid overlay network outlined in [10]. Moreover, the described scenario is currently being realized in a simplified version in the course of the research project SmartProducts. Results of experiments conducted with this prototype will be used to verify and enhance the presented configuration of the simulation model.

Acknowledgments. Part of this research has been funded under the EC 7th Framework Programme, in the context of the SmartProducts project (231204).

References

[1] Mühlhäuser, M.: Smart Products: An Introduction. In: Mühlhäuser, M., et al. (eds.) Aml 2007 Workshops. CCIS, vol. 11, pp. 158–164. Springer, Heidelberg (2008)

[2] Ahulló, J.P., López, P.G., Artigas, M.S., Arias, M.A., Aixalà, G.P., Bruchmann, M.: PlanetSim: An extensible framework for overlay network and services simulations. Architecture and Telematic Services Research Group, Technical Report DEIM-RR-08-002 (2008)

[3] Dabek, F., Zhao, B., Druschel, P., Kubiatowicz, J., Stoica, I.: Towards a Common API for Structured Peer-to-Peer Overlays. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 33–44. Springer, Heidelberg (2003)

[4] Law, A.M.: Simulation Modeling and Analysis, 4th edn. Mcgraw-Hill Professional (2006)

[5] Miche, M., Ständer, M., Brogle, M.: Leveraging Process Models to Optimize Content Placement - An Active Replication Strategy for Smart Products. In: 5th ERCIM Workshop on eMobility, Vilanova i la Geltrú, Spain, pp. 27–38 (2011)

[6] Kangasharju, J., Schmidt, U., Bradler, D., Schröder-Bernhardi, J.: ChunkSim: simulating peer-to-peer content distribution. In: Proceedings of the 2007 Spring Simulation Multiconference, Norfolk, Virginia, vol. 1, pp. 25–32 (2007)

[7] Steinmetz, R., Wehrle, K.: Peer-to-Peer Systems and Applications. Springer-Verlag New York, Inc. (2005)

[8] Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509 (1999)

[9] Hugues, P., Golenzer, J.: A Virtual Plane to Build and Maintain Real Ones. SmartProducts Whitepaper (March 2010)

[10] Miche, M., Erlenbusch, V., Allocca, C., Nikolov, A., Mascolo, J.E., Golenzer, J.: D4.1.3: Final Concept for Storing, Distributing, and Maintaining Proactive Knowledge Securely. EU FP7 SmartProducts, Deliverable (2011)

[11] Kröner, A., Haupert, J., Brandherm, B., Miche, M., Barthel, R.: Towards a Model of Object Memory Links. In: International Workshop on Networking and Object Memories for the Internet of Things, Beijing, China (2011)

[12] Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: An approach to universal topology generation. In: MASCOTS, p. 0346 (2001)