Abstract
This paper presents a Scenario-Based Adaptive Service Framework, where we can provide more adaptive service scenarios among highly correlated business applications. Individual services are seamlessly integrated into a framework by registering scripts, written in XML infrastructure language, called the XLogic, to the middleware platform, the XCREAM (XLogic Collaborative RFID/USN-enabled Adaptive Middleware). The XLogic (eXtensible Logic) script describes specific service scenarios depending on the events such as automatic identification or remote measurement data from the separate RFID/USN middleware.

1. Introduction
As advancements in computer and network technologies sped up, the technologies have rapidly been applied to a wide range of the real world. Especially, telecommunication technology has been continuously evolved during the past decades, and then incorporated with the traditional equipments including ordinary outside plant facilities, measurement devices, diverse sensors, various mobile devices and even motor vehicles. The communication ability given to those equipments implies boundless potential power, because we can always monitor the current status of the objects and cope with unexpected situations. Moreover, from the networking points of view, the smart objects equipped with communication devices are reachable by the correspondent parties at any time, anywhere and consequently, play their essential roles in constructing highly networked ubiquitous computing environment.

The ubiquitous computing environment allows us to take advantage of the benefit of the well-defined, structured services and continue to develop far more complex services by combining and enhancing the individual services. For instance, each physical entity equipped with RFID (Radio Frequency Identification) or USN (Ubiquitous Sensor Network) communication equipments is originally deployed to perform its own specific functions such as automatic identification or remote measurement. In addition, many other application services may share the meaningful information with each other to support the collaborative service.

Various services including emergency rescue system, smart facility management system, intelligent traffic system, surveillance system, logistics system, and supply chain system are not only introduced for their own purposes, but also interests in making organic connections or integration of existing independent application services have been increasing greatly. Especially, the increasing desire for the better pleasant and healthy environment accelerates the collaboration of those individual services. The more application services are introduced, the higher the rate of collaboration of the application services grows.

In order to provide complex services of organically connected applications in a RFID/USN equipped environment, a robust and stable infrastructure is necessary to support the reliable collection of a large amount of identification and sensor data, and real-time delivery of the data according to a predefined scenario of individual services. As a solution for this requirement, a scenario-based collaborative middleware platform needs to be implemented as a core infrastructure system, which seamlessly interfaces highly sophisticated services within the RFID/USN based collaborative framework.

As the fundamental infrastructure of the framework, we have developed the XCREAM (XLogic Collaborative RFID/USN-Enabled Adaptive Middleware) platform, which plays an organizing role for ultimate ubiquitous environment and maximize the collaboration between the services.

The application of the framework seems to be rapidly increasing and expanding in our day-to-day life, as we require highly sophisticated services, which is triggered by a specific event from a variety of sensors or other services according to a pre-defined scenario. Intuitively, this is accompanied by frequently seamless integration between many service systems, which have been operated independently by registering scenario-based scripts written in XML infrastructure language, called the
“XLogic (eXtensible Logic)”, to the middleware platform, the XCREAM. The XLogic script describes specific service scenarios depending on the events such as automatic identification or remote measurement data from the separate RFID/USN middleware.

2. Related research

There have been many researches related to the proposed system: RFID-enabled auto-identification technology for the data collection and recognition of the front-end component of the framework; massive event handling technology for the internal events processing of the XCREAM; and USN middleware technology, which many similar systems adopted in the fields [19,20].

In conjunction with handling RFID tag data, a lot of research efforts have been made in the last decade to define the RFID tag specification. Research has also been made to formalize the protocols required for seamless communication between related components and to enhance the performance and correctness of the system itself.

Initially, RFID-enabled network infrastructure was proposed by the Auto-ID Center [21]. Their research goal is to realize ubiquitous automated identification technologies based on the networked physical world [7]. They have developed an open architecture system which is composed of the EPC, EPC tags and readers, local networking technology, and RFID middleware [26,27]. Local networking technology allows readers and sensors to be connected via local databases. RFID middleware collects and filters massive tag data, aggregates and processes them into meaningful information, and delivers them to the pertinent applications. The middleware may use the Object Name Service (ONS) similar to the Domain Name Service (DNS) in the Internet for location lookups of specific items [6].

A variety of automatic identification devices under highly-networked environment generate a huge amount of events streams. This means that a new querying scheme is required to recognize the arrival of the corresponding events and to evaluate the corresponding queries. In order to do so, users need to issue a continuous query (CQ) which is invoked once and run continuously over the events streams and database, and notified whenever the data matches the query [22]. The continuous query processing system must be capable of monitoring the incoming events that occur during certain predefined time intervals and generate new results when they become available [4,5].

Moreover, the event queries are distinguished from the traditional queries that carry over the relations by many researchers who analyze the events and the nature of the event queries [1,2,15,17,24].

Many data stream management systems (DSMS), such as TelegraphCQ [4]; NiagaraCQ [5]; STREAM [30] and OpenCQ [13], have been developed as solutions to handle multiple continuous, high-volume, and possible time-varying data streams. The TelegraphCQ adopts an adaptive query engine to process queries efficiently in volatile and unpredictable environments. The NiagaraCQ keeps scalability by grouping CQs. The STREAM maintains adaptive and dynamic central scheduler of query operators over shared stream queue which holds high-volume and bursty traffic of data. The OpenCQ focuses on a query processing algorithm based on the incremental view maintenance [2].

These systems are primarily focusing on CQ optimization by allowing traditional DBMS, or their own Data Stream Management System (DSMS), to support CQs over data streams and conventional relations. The XCREAM supports CQs over time-varying RFID tags or sensor data by caching the runnable objects of the XLogic scripts correspondent to them.

The requirements of the early USN middleware were comparatively simple due to the direct connection to a specific application service. As various kinds of application services, such as u-Healthcare, u-Transportation, u-911, and u-Eco based on ubiquitous environment are rapidly introduced to our day to day life, it is necessary that the information from many RFID/USN devices interrelates with each other [9,10,11]. Several middleware researches are having conducted on the following examples: MiLAN middleware [8], developed to guarantee QoS (Quality of Service) as its top priority; event-based DSWare middleware [12], which sends desired data to USN application systems by setting events on continuous stream of sensor data obtained, similar to RFID middleware; Impala middleware [12,14], which can dynamically change functions of sensor node middleware according to changes in USN application services and changes in the surrounding environment of sensor networks through wireless communication; TinyDB middleware [16] and Cougar middleware [23], which carries out requirements of USN application system using distributed processing by considering sensor data from the sensor network as distributed data of distributed database.

In COSMOS (COmmon System for Middleware Of Sensor network), developed by ETRI, key functions of middleware, jointly needed for various types of USN application services, were extracted, and technology development and standardization to provide these in a standardized way were carried out. The main functions of COSMOS are as follows: query support (temporary, permanent, and event); support for simultaneous processing of a large amount of queries for large-capacity sensor network environments; and support for abstraction of heterogeneous sensor networks [18,19].
Oracle BPEL (Business Process Execution Language) Process Manager, developed by Oracle, agrees with the SOA concept, and provides a method of integrating highly distributed heterogeneous application infrastructures [3]. Oracle BPEL Process Manager makes loose coupling possible by having application services expose their services to the external world by using WSDL (Web Services Description Language) and by connecting services from heterogeneous environments by using HTTP-protocol-based SOAP (Simple Object Access Protocol). Also, business processes among heterogeneous services can be defined through Oracle Business Process Execution Language.

3. XCREAM framework

The XCREAM provides a scenario-based collaborative framework, which seamlessly integrates various application services in the RFID/USN environment. Within the XCREAM, the Agent Manager plays a bridge role between the various application services and physical RFID/USN Middleware systems. The Agent Manager maintains the runnable objects of the XLogic scripts, which invoke the corresponding web services of the application services. Upper-level application services are to collaborate with each other, through the Agent Manager, which allows pre-registered scripts to be executed depending on the event data collected by the RFID/USN Middleware systems. The scripts are written in XML infrastructure language, called the XLogic, which describes specific service scenarios and interact with the middleware platform. Each agent may exchange event data, under the supervision of the Agent Manager. Their associated relationships are shown in the figure 1.

The XCREAM adopts the service-oriented architecture (SOA) as an interfacing scheme for integrating RFID/USN-based application services not only to guarantee independence of individual services, but also to gain flexibility when extending the platform infrastructure by enabling new collaborative services to work with the platform or other application services. Basically, the SOA works through the following steps: existing application functions are classified into functional business units; they are re-combined as software component units, namely, services, through standardized calling interfaces; and the services are combined in order to construct the real-world business applications [31,32]. Based on the SOA structure, the platform exposes customized web services as the method of establishing connections with heterogeneous external systems.

The XCREAM platform is based on the event-driven architecture (EDA), which is completely decoupled from the specific RFID/USN middleware. This scheme enables the platform to work independently with an RFID/USN middleware and execute the scenarios only when the relevant events from the RFID/USN middleware are recognized. To promote the portability of the middleware platform, it is written in Java programming language [25,28,29]. In addition, a script language called XLogic allows users to register the scenarios, which are activated by a specific event and deliver the event to the related service applications. In order to support real-time processing, a caching system is also considered, which may keep XLogic scripts within memory, depending on their frequency of use. This scheme is expected to shorten the time for parsing the scripts.

4. XLogic and Scenario-based collaboration

The development of the XCREAM was initiated to present flexible collaborative framework for the various kinds of information systems, which have not only been used in many independent sites, but also newly introduced RFID/USN-enabled application services. In order to fully support the original philosophy, we had to devise a completely seamless interfacing scheme to the 3rd party applications and their associated RFID/USN middleware systems. This motivation enabled us to design an XML based infrastructure scheme, the XLogic script language, as a base interface method, and to seamlessly integrate the application services and their related RFID/USN middleware systems to the front-end of the XCREAM platform. Moreover, many services are to collaborate by obtaining RFID tag information or any

![Figure 1. The XCREAM Framework](image-url)
other sensing data and applying them to their collaborative scenarios.

4.1. The XLogic script language

The XLogic script language is used to define a specific service scenario for each application service, which is registered in the XLogic Scripts Repository of the XCREAM framework as shown in the figure 1. By specifying the name of the related script in the RFID/USN middleware, individual events from the physical identification or sensor equipments interconnected to the middleware are to trigger the corresponding XLogic scripts.

The XLogic script language follows the XML-based scheme and provides the application services with the following statements in the form of the XML tags:

- set statement
- wait statement
- print statement
- if statement
- while statement
- foreach statement
- invokeWebService statement
- break statement
- continue statement

The following shows a sample XLogic script, which registers a service called “PlayerInfoService” with the final destination of the tag data to be sent to the specified url.

```xml
<xlogic:invokeWebService
  service="PlayerInfoService"
  url="http://baseballteam.bcity.com/pm/baseball"
  name="find_player">
  <rawtags xmlns="bcity">
    <xlogic:iterator name="tag" source="tags">
      <tag>${tag}</tag>
    </xlogic:iterator>
  </rawtags>
</xlogic:invokeWebService>

<xlogic:set name="player"
  select="/id">${find_player}
</xlogic:set>
```

4.2. The XLogic execution

The XLogic script is registered to the XCREAM platform in either trigger mode or immediate mode. An XLogic script, which is defined as the trigger mode, is executed when the external event is collected by the XCREAM platform according to the flow of event depicted in the figure 2.

![Trigger Mode](image)

The immediate mode refers to the mode in which an XLogic script written by a user is transmitted into the XCREAM Enterprise Manager and then executed immediately as shown in the figure 3.

![Immediate Mode](image)

5. Testbed environment

In this section, we briefly introduce our testbed environment of the XCREAM framework, which is configured in Auto-ID Laboratory in Towson University.

First of all, the testbed environment includes two RFID readers with two pairs of antennas as well as three wireless temperature and humidity sensors as shown in the figure 4.

In the testbed environment, the XCREAM server is connected with the RFID Reader and the data collector of the wireless temperature and humidity sensors. This testbed works as a monitor for automatic identification of
Gen 2 Standard RFID tags and abnormal temperature and humidity alarms.

As the collaboration infrastructure, many application services can be combined together to provide more sophisticated services which take advantage of each service’s basic functionalities.

![Image](image_url)

**Figure 4. Testbed Environment**

6. Future expectations

The introduction of the XCREAM based approach is expected to increase the user needs to attach a variety of innovative smart devices and the associated services to the existing service framework with elaborate collaborative scenarios.

Improved services with a variety of combination of individual services would remarkably change our day-to-day life a lot more comfortable and safer than before.

The wide adoption of the RFID/USN-enabled application services with the proposed framework accelerates rapid prototyping and deployment of the new services with the effective customization methods.

The developed XCREAM framework in this paper can make a cooperative work possible among existing middleware systems and enable a variety of RFID/USN-enabled services to be combined within the flexible framework to provide composite and refined services.

Furthermore, as the XCREAM framework has not only focused on seamless integration of the external services, but also the autonomy of the individual services, this framework would be well suited to the complex RFID/USN-enabled command and control center solution of the “Smart City.”

7. Conclusion and future works

This paper focuses on the implementation of the collaborative XCREAM framework with the specification and the execution process of the XLogic script language which is used to define either individual or collaborative service scenarios. In addition, the testbed environment configured with the RFID tag readers and wireless temperature and humidity sensor equipments is presented. Based on this framework, our research team continues to devise many effective solutions with well-organized service scenarios.

So far, the research team has tested the required functionalities of the XCREAM platform, itself and the performance of the framework by combining test services.

In the near future, the benchmark test of the XCREAM platform by comparing it with other solutions as well as the application of the framework to the real world cases would be done.

8. References


