

A Process Framework for Semantics-aware Tourism Information Systems

Olawande J. Daramola

Department of Computer and Information Sciences,
Covenant University, Ota, Nigeria.
jodaramola@covenantuniversity.com

Abstract: The growing sophistication of user requirements in tourism due to the advent of new technologies such as the Semantic Web and mobile computing has imposed new possibilities for improved intelligence in Tourism Information Systems (TIS). Traditional software engineering and web engineering approaches cannot suffice, hence the need to find new product development approaches that would sufficiently enable the next generation of TIS. The next generation of TIS are expected among other things to: enable semantics-based information processing, exhibit natural language capabilities, facilitate inter-organization exchange of information in a seamless way, and evolve proactively in tandem with dynamic user requirements. In this paper, a product development approach called Product Line for Ontology-based Semantics-Aware Tourism Information Systems (PLOSATIS) which is a novel hybridization of software product line engineering, and Semantic Web engineering concepts is proposed. PLOSATIS is presented as potentially effective, predictable and amenable to software process improvement initiatives.

Keywords: *Product line, Web engineering, Software Process Improvement, Tourism Information System, Semantic Web, Software product development.*

1 Introduction

The advent of new technologies has promoted thoughts of new intelligent possibilities in Tourism Information Systems (TIS) which has also amplified the complexity of the tourism requirements landscape. According to [1], as a result of emerging new technologies like the Semantic Web and mobile computing, the modern tourist is prone to a number of dynamic characteristics, which are: 1) become more mobile and critical; 2) become less loyal and frequently change their product preferences; 3) look for more specialized products and ask for better services; 4) want more and better information; 5) compare more products in more detail; 6) have fast changing needs and belong to different niches at the same time; and 7) tend to make more but shorter vacations. The dynamic characteristics of tourism consumers' behaviour have aggravated the complexity of functional requirements of TIS which portend a critical challenge for providers of e-tourism services [1], [2].

Although a number of attempts have been made to tackle this problem, none of the approaches so far reported in literature has been centred on a thinking that seeks to solve this problem from a software development process perspective. This scenario

raises a pertinent research question: 1) how can developers of TIS effectively respond to the trends in consumer behaviour from a product development perspective? Or in another form: Is there a product development approach or methodology that could be engaged to tackle dynamic requirements in tourism? This is the thinking that must guide the development of the next generation of TIS.

In this paper, a software process framework called Product Line for Ontology-based Semantics-Aware Tourism Information Systems (PLOSATIS) is proposed as a novel hybridization of software product line engineering, ontology engineering and semantic computing for developing next generation TIS. Specifically, the next generation of TIS are expected among other things to: enable semantics-based information processing, exhibit natural language capabilities, facilitate inter-organization exchange of information in a seamless way, and evolve proactively in tandem with dynamic user requirements [2], [3]. PLOSATIS leverages concepts and practises from ontology development because ontology is a fundamental component for achieving the Semantic Web. Ontology has the capability to solve a number of problems in tourism. This includes: 1) enabling interoperability of heterogeneous platforms; 2) standardization of business models, business processes, and knowledge architectures; and 3) serving as a model of knowledge representation for the generation of knowledge-based information services [4].

PLOSATIS also leverages the Software Product Line Engineering (SPLE) paradigm because of SPLE's ability to engender systematic and strategic reuse of software artifacts based on identifiable common domain requirements so as to facilitate rapid market-entry and flexible response to dynamic user requirements [5]. Hence, SPLE-based PLOSATIS is designed not only to enable the production of next generation TIS product families with semantic-awareness, but also, to facilitate the proactive evolution of such TIS in response to dynamic user requirements. Another vital aspect of PLOSATIS is semantic computing, because of the need to equip next generation of TIS with semantic processing capabilities that will engender meaning-based execution of information retrieval and information processing tasks such as semantic searching, semantic querying, and semantic browsing unlike conventional dumb portals that are currently prevalent in the e-tourism landscape [6]. Hence, PLOSATIS is designed to enable the development of TIS that will exhibit Semantic Web and semantics-aware information processing capabilities that is mostly non-existent in many e-tourism platforms.

The rest of the paper is organised as follows: Section 2 is a review of related work from literature, while Section 3 describes the PLOSATIS framework in detail. In section 4, an experimental application of PLOSATIS is presented and the results obtained, also a discussion on the preliminary assessment framework used to evaluate PLOSATIS is given. In Section 5 the paper is concluded with a brief note.

2. Related Work

Thus far, relatively few approaches for modeling of Semantic Web applications have been reported in literature. Some of these include: Semantic Hypermedia Design

Method (SHDM) [7], [8] which is a model-driven approach to designing web applications using five steps: Requirements Gathering, Conceptual Design, Navigational Design, Abstract Interface Design and Implementation. SHDM also embraces the use of ontological definition languages such as DAML+OIL and OWL, for expressing advanced aspects such as constraints (restrictions), enumeration and XML Schema datatypes. The OntoWebber system [9] is an ontology-based approach to website management. It facilitates the design, creation, generation and maintenance of Web sites using a set of software tools. It also enables the personalization of Web site views based on individual users. Another notable approach is the Hera project [10], which is a methodology that supports the design and engineering of Web Information Systems (WIS) using Semantic Web technology. The main focus of the Hera project is to support Web design and implementation particularly hypermedia aspects.

However, these aforementioned approaches are quite generic and do not address the peculiarities of specific application domains. Also, although they leverage Semantic Web technology to exhibit some forms of intelligence (personalization - OntoWebber, hypermedia presentation – Hera), they were not setup primarily to facilitate semantic-awareness in WIS as envisioned by PLOSATIS. Additionally, PLOSATIS, unlike others, is based on a product line perspective in order to facilitate proactive evolution of TIS (which are specializations of WIS for tourism).

3. Description of the PLOSATIS Framework

PLOSATIS is more amenable to a context where the following obtains: 1) All developed TIS belong to the same organization or a consortium of collaborating organizations; 2) the requirements of different variants of TIS products within the product family are well known and can be predetermined in advance; 3) the process description and tools for developing specific kinds of TIS product can also be predetermined. In essence PLOSATIS fits more into the context of a software developer organization.

- *The PLOSATIS Lifecycle*: The flow of activities in PLOSATIS (see Figure 1) is sequential, but also iterative. The breakdown of the activity workflow is as follows:
 - i) *Requirements and Process Analysis*: This consist of activities that provide the necessary managerial guide and organizational control that complements the technical aspects of ontology engineering, domain engineering, and Web product engineering of the PLOSATIS framework. Its main sub-activities are as shown in Figure 1. *Configuration management* ensures that changes that need to be made to products by way of upgrade and versioning are carefully planned in a way that makes them technically realizable without disrupting the design of the product line, while *critical evaluation* ensures that periodical evaluations are carried out at specific points, notably after ontology engineering, domain engineering and Web product engineering to determine whether the process should proceed based on cost incurred so far and rate of productivity.
 - ii) *Ontology engineering*: This comprises ontology modelling, ontology design, and ontology development activities, that are required for the realization of the

reusable knowledge artifacts needed for the execution of the PLOSATIS process.

- iii) *Domain engineering*: During domain engineering, the reference architecture for the product line is created that consists of the core components (reusable software assets that are needed for building the variant TIS products) of the product line. The components are constructed from scratch or sourced as Commercial off the Shelves (COTS) components, which are then tested and certified for reuse in the product line. Also, semantic computing capabilities are introduced into the domain components using appropriate middleware tools and technologies, or through the implementation of appropriate algorithms for natural language understanding. The key sub-activities of domain engineering are: domain analysis, domain design, domain realization and domain testing.
- iv) *Web product engineering*: This is concerned with the creation of specific TIS products through the reuse of core assets created in domain engineering. The operations at this level are typically the weaving together of customized domain components and the creation of hypermedia contents. The focus of Web product engineering is more toward integration and customization rather than construction based on reuse of existing domain components. Semantic capabilities can also be introduced at this level depending on the peculiarity of functionality required by a product. The core activities of product engineering are product analysis, product design, product realization, and product testing.

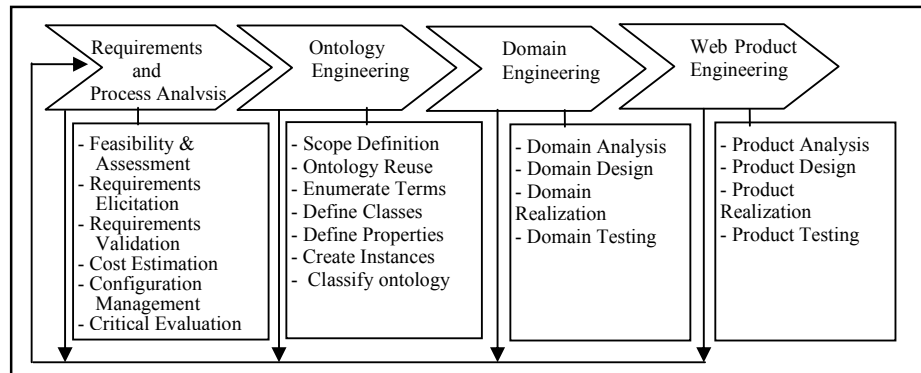


Fig. 1. PLOSATIS process Lifecycle

4. Experimenting with PLOSATIS

An empirical study with the PLOSATIS framework is being undertaken by the SEAI group of Covenant University, Nigeria. This entailed the creation of a prototype National Tourism Semantic Web Portal (NTSP) in order to archive, and provide web-access to Nigerian tourism data in a much more meaningful via the web and mobile platforms.

- *Requirements and Process Analysis in PLOSATIS*: PLOSATIS was initiated with the requirements and process analysis activity which probed into the expected

functional characteristics of the NTSP. Important domain components and relevant ontologies for the NTSP were identified at this stage.

- *Ontology Engineering in PLOSATIS*: A suite of ontologies was developed to serve as the semantic backbone for prospective products of the PLOSATIS framework. Specifically, four tourism OWL ontologies were developed from scratch. This was preferred to using existing tourism ontologies in order to evolve customized ontologies that comprehensively model the Nigerian tourism terrain, such that specific data on several tourism aspects such as destination, accommodation, restaurant, events, travel that pertain to Nigeria can be captured. The ontologies were designed to store contextual information about tourist locations and their operational time characteristics such that the high mobility of the modern tourist and the dynamic changes that can occur during a trip can be catered for, thus enabling the generation of personalized and context-aware travel information services. All the ontologies were implemented as OWL ontologies using the Protégé Ontology development tool. The ontologies are:

i) *Nigeria Destination Ontology (NDO)*: The NDO is a semantic representation of information on specific attributes of Nigerian destinations. The concept of a destination is conceived to consist of three types: *City*, *Town* and *Village*; each destination type has six attributes namely: *Tourism Asset*, *Weather Temperature*, *Scenery*, *Volume of Traffic*, *Crime Rate*, and *Status*. Also, Villages and Towns are conceived as extensions of specific City destinations and thus are related to specific destination subclasses instances using ‘*PartOf*’ association. *Tourism Asset* is an enumeration of the types of tourism artifacts that are available in a destination (10 different categories were identified i.e. beaches, landforms, conservation, recreational parks etc.); each category has equivalent *datatype* property (e.g. *hasLandforms*, *hasBeaches* etc.) which is used to capture the details (full name and location address) of all tourism assets belonging to that type (a *maximum cardinality* of 25 is imposed). Thus the NDO models specific destinations as an agglomeration of available tourism assets, and associated contextual information.

ii) *Nigeria Accommodation Ontology (NAO)*: The NAO is a semantic representation of the attributes of the various types of tourism accommodation. Seven specific attributes of accommodation types (e.g. hotel, guest house, hostel, chalet etc.) were considered. These are 1) *Services* - the description of kinds of services rendered; 2) *Gastro* - profile of eateries, cuisines or restaurant nearby; 3) *Attraction*- special attractions within or nearby; 4) *State*: province or region where it is located; 5) *Facilities* - physical facilities available; 6) *Location*- specific location of the resource and 7) *Time* – the working time period of the establishment.

iii) *Nigerian Events Ontology (NEO)*: The NEO was created to capture information about the events that are considered significant for Nigerian tourism. The NEO consists of five disjoint subclasses namely: *Location*, *State*, *Sponsor* (the organization that sponsors the event), *Time* (month of the year), and *Event* (a classification of the event into specific event category). Seven types of event category were identified. These are: Cultural, Social, Educational, National, Political, Religious, and Sports.

- iii) *Nigerian Restaurants Ontology (NRO)*: The NRO is a semantic representation of information on Nigerian restaurants. The NRO consists of four disjointed classes namely: *Restaurant, State, Location, and Time*. The class *Restaurant* was defined as consisting of 26 subclasses which correspond to 26 standard types of restaurants to which a typical restaurant can belong. All the ontologies import the 3WC Time [11] and the 3WC Geo [12] ontologies in order to model user's time and location context.
- iv) *Nigerian Travel Ontology (NTO)*: The NTO is an integral part of the NTSP architecture though it is not yet implemented. It is conceived as a composition of all important knowledge artifacts relevant to the Nigerian travel industry. Aside importing the NDO, NAO, NEO and NRO, the NTO will also contain knowledge about other tourism objects such as transportation (land, and local air travel).
- *Domain Engineering in PLOSATIS*: The domain engineering activities entails the identification, design, and implementation of the core components of the National Tourism Semantic Portal (NTSP) architecture. The NTSP has a layered architecture consisting of the five layers (See Figure 2). The layers are described as follows:
 - i) *Client Layer*: This layer is comprised of client devices through which the services of the NTSP are requested. Typical clients include PDAs, web browsers (through Laptop and PC), Web services and Mash-up applications.
 - ii) *Protocol Layer*: This layer defines the implementation platform for all the services in the NTSP. This can be WAP for WAP-enabled mobile applications, SMS-based applications or HTTP for Web clients. This layer is responsible for the determination of appropriate communication protocol depending on the nature of requesting client and the provision of context-aware services.
 - iii) *Portal Services Layer*: The portal services layer consists of the class of all services that are available in the NTSP architecture. These services are rendered by specific domain components that leverage semantic information stored in the underlining ontology layer. Services can be broadly classified as 1) information services such as semantic query processing, semantic searching, and semantic browsing; and 2) Transaction services which are basically recommendation services rendered by different recommender systems, for destination, accommodation, restaurant, and events.
 - iv) *Semantic Middleware Layer*: This layer encapsulates the set of middleware infrastructures that enables the NTSP components with Semantic Web (to read, and process facts stored in ontologies) capabilities, and enables semantics-aware processing of user queries. Relevant middleware components, tools, and algorithms (such as: stemming, part-of-speech tagging, content summarization, and query reformulation, data mining) are deployed at this layer in order to trigger semantic reasoning, semantic searching, semantic mining, semantic browsing, and semantics-based recommendation services that are available on portal services layer of the NTSP.
 - v) *Data and Ontology Layer*: This layer represents the semantic backbone for all services in the architecture. The components of this layer are the suite of knowledge representation ontologies that are designed to facilitate intelligent

personalized services in the NTSP. This layer also contains the set of database abstractions that stores the tourism information contents that are exploited by the NTSP.

- Implementation of Domain Components:** The implementations of domain components were based on Java technology, using the Jena Semantic Web development framework and the NetBeans 6.5 Java IDE. The Sun Application Web Server 9.0 was used as middleware for deploying server components. The Web GUI and functionalities were implemented using Macro Media Flash and Dream Weaver web design tools, and Java Server Pages (JSP). The recommender systems were implemented as Enterprise Java Beans (EJB) components embedded in the web interface. Each of the EJBs references the specific ontology to which they were mapped using the Jena ontology APIs [13] to trigger ontology querying and reasoning capabilities. The Pellet 2.0 Descriptive Logics (DL) reasoner [14] was used as the reasoning engine for the ontologies, while JESS was used as the rule engine. All the recommender system components in the NTSP are knowledge-based RS. Each of the ontology in the NTSP is mapped to a corresponding recommender system, the recommender systems then generates list of top nearest neighbourhood recommendations using a content-based filtering approach. So far two of these recommender systems have been realized which are the destination recommender and the Accommodation recommender system. The other recommender systems are scheduled to be realized in the later phases of the project, So far, a first prototype of the NTSP has been built which is an extension of the work reported in [4].

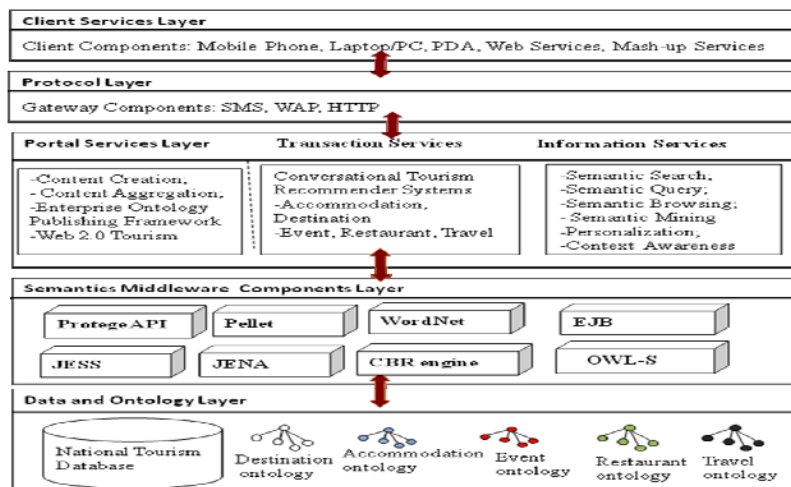


Fig. 2. A layered schematic architecture of the NTSP

- Web Product Engineering in PLOSATIS:** During this phase, the domain components were customized with specific functionalities, ontologies populated with specific data instances, and creation of hypermedia contents (web graphics,

animations, sound etc.) in order to realize the TIS products based on the feature model of the product line. The feature oriented domain analysis method (FODA) [15] was used. Based on the feature model, the destination recommender, accommodation recommender, and semantic query engine are the mandatory components in TIS product line, while other components are optional. Thus far the compulsory components have been realized in the first prototype of the NTSP.

4.1 Evaluation Framework for PLOSATIS

Three of the seven key goals of a software process as proposed in [16] are considered as most relevant to the context of PLOSATIS as a proposed software process. These are *effectiveness*, *predictability*, and *improvement*. Hence the choice to evaluate PLOSATIS along these three dimensions.

- *Determining Effectiveness of PLOSATIS*: The effectiveness of a software process is solely determined by its ability to deliver the right product that is, the extent to which its output is valuable to the user. A potentially viable approach for evaluating the effectiveness of PLOSATIS is the Goal Question Metric (GQM) paradigm [17]. The GQM is a kind of software measurement that offers a systematic approach for tailoring and integrating goals with models of software processes, products and quality perspectives of interest, based on the specific needs of an organisation [17]. It is a 3-level process comprising of: the identification of the key goals of a process, the formulation of goal-specific assessment questions, and the definition of metrics to generate data to answer the goal-specific questions in a quantitative way. For instance, in the NTSP project, the adopted GQM model to determine the effectiveness of PLOSATIS is presented as follows:

The PLOSATIS Effectiveness Model:

Let:

$M1_c$ = user rating for M1; $M2_c$ = user rating for M2; $M3_c$ = user rating for M3;

$M4_c$ = Confidence Factor; $M5_c$ = user rating for M4

Let $\{PQ_i\}$ be set of Weighted Product Quality per User [each $PQ_n = (M1_c * M4_c * M5_c)$]

Where Maximum $PQ_n = 125$ [for a perfect user rating], $n \leq i$

Let $\{AF_i\}$ be set of Product Appreciation Factor per User

[each $AF_n = [PQ_n / (M2_c * M3_c)]$]

Where Maximum $AF_n = 25$ [for a perfect user rating], $n \leq i$

Interpretation Algorithm for PLOSATIS Effectiveness Model

[Where 62.5 is a benchmark value that corresponds to average rating (PQ_n) of PLOSATIS by a user]

If 70% or more of $\{PQ_i\} \geq 62.5$ then PLOSATIS is perfectly effective process

If 60% - 70% of $\{PQ_i\} \geq 62.5$ then PLOSATIS is very effective process

If 50% - 60% of $\{PQ_i\} \geq 62.5$ then PLOSATIS is mostly effective process

If 45% - 50% of $\{PQ_i\} \geq 62.5$ then PLOSATIS is barely effective process

If less than 45% of $\{PQ_i\} \geq 62.5$ then PLOSATIS is not effective process

Figure 3 gives a brief overview of questions and metrics of data collection for the draft GQM.

<p><i>Question 1:</i> Can you characterize your experience with Tourism Information System (i.e. e-Tourism websites)?</p> <p><i>Metric M1:</i> Subjective rating per person [0-5]; 0 (lowest) – 5 (highest)</p> <p><i>Question 2:</i> How strong is your desire to obtain intelligence and semantic-awareness from Tourism portals?</p> <p><i>Metric M2:</i> Subjective rating per person [0-5]; 0 (lowest) – 5 (highest)</p> <p><i>Question 3:</i> Can you characterize your perception of the level of semantic-awareness of tourism portals based on experiences from the past?</p> <p><i>Metric M3:</i> Subjective rating per person [0-5]; 0 (lowest) – 5 (highest)</p> <p><i>Question 4:</i> What is your assessment of the quality of semantic-awareness of PLOSATIS products?</p> <p><i>Metric M4:</i> Confidence factor = (number of agreeable outcomes) / (number of trials)</p> <p><i>Metric M5:</i> Subjective rating per person [0-5]; 0 (lowest) – 5 (highest)</p>
--

Fig. 3. Questions and Metrics used for the GQM

Using the draft GQM model in Figure 3, a usability experiment can be undertaken using a population of qualified users of the system as participants to determine the effectiveness of the PLOSATIS process with respect to the specific problem scenario at hand. Questionnaire based on the developed GQM model can be given to each of the participants to capture their evaluations. The collected data could subsequently be analysed statistically to determine the measure of effectiveness of PLOSATIS. A first-cut GQM evaluation conducted by our group revealed PLOSATIS as a mostly effective software process (details not provided due to space constraint).

- *Determining the Predictability of PLOSATIS*: A software process is deemed predictable if it is possible to reasonably estimate the cost of developing products by using the process. According to [16], a good software process is one that layout clearly the steps of development such that planning for new products, and proper allocation of resources of both time and people, can be done ahead of time. PLOSATIS satisfies this requirement because it has clearly defined activities, from which specific cost parameters can be derived for project cost predictions. A cost estimation model that is based on augmentation of the SPL estimation model given by [18] has been adopted for PLOSATIS. The cost model which computes efforts in Person Months is given as follows:

$$E_{plosatis} = E_{req} + E_{onto} + E_{dom} + E_{ontoupdate} + N * (E_{reusewith} + E_{uniquewith} + J * E_{updatewith})$$

Where

E_{req} , E_{onto} , E_{dom} , $E_{ontoupdate}$ - are efforts expended on: requirements engineering, ontology engineering, domain engineering, and periodic update and maintenance of ontologies used in the project, respectively;

$E_{reusewith}$, $E_{uniquewith}$, $E_{updatewith}$ - are efforts expended on: Web product engineering for the reuse of existing core assets, the manual adaptations of core assets after creation, and the update of product-related core assets in the core asset base, and creation of additional core assets that are unique to a product respectively;

N - The number of TIS products in the product line; and

J - The average planned number of content update cycles for one TIS product.

E_{onto} and $E_{ontoupdate}$ can be estimated based on the ONTOCOM model as proposed in [19], while the other metrics of the estimation model can be based on COCOMO II [20]. This derived effort estimation model for PLOSATIS therefore provides a basis for a priori cost prediction of future projects using relevant historical data in the instance of the adoption of PLOSATIS by an organization.

- *Amenability to Software Process Improvement of PLOSATIS*: Amenability to software process measurement and improvement is a vital quality of a good software process. However, the hybrid nature of PLOSATIS ensures that none of the existing software assessment frameworks such as: SPICE, CMMI, BOOSTRAP and Goal Question Metric (GQM) [21] is perfectly suitable for its assessment as a software process. Therefore a 32-questions template that spans the four aspects of the PLOSATIS development lifecycle has been formulated which could be used as basis for Software Process Improvement (SPI) initiative. The questions (see Table 1) were based on best-known practices and rules that govern the conduct of software product line initiatives [21], Ontology engineering activities [4], and Web engineering [7], [8], [10].

Table 1: Questions Template for Software Process Assessment of PLOSATIS

Requirements and Process Analysis	
Question 1.	Does the organization's management in support of the pursuit of the software product line initiative and subscribe to its principles?
Question 2.	Is the influence of the management team on the execution and overall success of the software product line initiative considered significant?
Question 3.	Is adequate risk assessment undertaken in order to compare the expected investment with accruable benefits that can be gained from the pursuit of software product line before commencement of projects?
Question 4.	Is a comprehensive domain analysis undertaken prior to product development in the software product line?
Question 5.	Is there a requirements engineering scheme for the software product line that ensures that the product line requirements are well documented, defined, analyzed, verified and managed?
Question 6.	Is there a reference architecture that fully captures the base requirements of the software product line and clearly identify the possible variability points in the software product line outputs.
Question 7.	Is there a configuration management scheme in place to address the configuration management issues that arise in the software product line?
Question 8.	Is a cost-benefit analysis conducted at the end of each software product line project to evaluate the pay-off of such efforts?
Question 9.	Is the cost benefit ratio or ROI of the software product line in agreement with the organization's financial projections?
Question 10.	Are the integral activities of software product line development executed iteratively?
Ontology Engineering Assessment	
Question 11.	Is the ontology engineering effort intensive and considered highly relevant?
Question 12.	Is the schedule for ontology development well documented?
Question 13.	To what extent are scheduled ontology development tasks executed as planned?
Question 14.	Is the conceptualization of the ontology in line with the specification of the product line architecture?
Question 15.	Are all ontologies consistent with the scope of the software product line?
Question 16.	Are adequate provision made for the timely update of the ontology with a documentation of update schedules?
Question 17.	Does the product line scope allow for the addition of new ontologies or integration of existing ontologies with other relevant ontologies as requirements evolve?
Question 18.	Are there provisions for ontology support activities such as acquisition, evaluation, integration, merging, and alignment?
Domain Engineering	
Question 19.	Are all of the domain components within the software product line core asset base and the eventual TIS products consistent with the software product line scope?
Question 20.	Are the variability points in all domain components in the core asset base well defined such that they can be properly customized for variant TIS products composition?
Question 21.	Is there a scheme in place to ensure that the core asset base gets updated regularly through the addition of new domain components as the TIS product line evolves?
Question 22.	Is there a version control management system in place to keep track of the domain components development and reuse history?
Question 23.	Do all the third party components (COTS) present or added into core asset base satisfy the cost benefit ratio for the organization?
Question 24.	Is the degree of domain components reuse very high relative to the number of TIS products developed within the software product line?
Web Product Engineering	
Question 25.	Is there a reference architecture that is shared by all TIS products within the software product line?
Question 26.	Is the degree of commonality among TIS products generated by the software product line very high?
Question 27.	Does the variation among TIS products well-known ahead of time and fully accommodated within the definition of the software product line scope?
Question 28.	Does every TIS product generated from the software product line represent a valid business case for the organization?
Question 29.	Does the software product line produce a many variant TIS products, or at least more two?

Question 30.	Is there sufficient automated support for customization activities and domain component integration activities to realize variant TIS?
Question 31.	Is there support automated tool support for hypermedia creation activities that would aid navigation and presentation of information in TIS products?
Question 32.	Does every TIS product released from the software product line certified as suitable by the organization?

The formulated assessment template (see Table 1) though not exhaustive could provide adequate basis for the application of standard SPI frameworks such as SPICE, CMMI, or GQM to PLOSATIS. For example, using the questions of the assessment template, coupled with the definition of appropriate goal-specific metrics, relevant quantitative data can be generated that would facilitate successful GQM evaluation of PLOSATIS along specific software process quality dimensions for improvement. Hence, PLOSATIS is quite amenable to software process improvement in the instance of its adoption by a TIS developer organization.

4.2 Discussions

The experience so far with PLOSATIS has revealed its potential to enable proactive evolution of TIS in tandem with emerging dynamic requirements. For example, the first set of prototypes that have been implemented does not yet cover the full scope of the reference architecture of the TIS product line, but there remain sufficient opportunities for future extensions and products evolution based on the predefined versioning scheme, which is to revise the products every two years. In order to realize other variants of the existing TIS product all that will be required is to add one or more of optional domain components that were already specified in the reference architecture. While mobile versions of all TIS products can be produced by customizing some of the exiting domain components for the mobile platform. Also, other future additions could be made to the reference architecture based on the dynamics of user requirements within this specific domain which will in turn provide a basis for new product variants that possess additional features and more advanced functionalities to evolve.

5. CONCLUSION

In this work, the PLOSATIS process framework has been proposed as a novel and viable software process for the development of semantics-aware TIS. It offers a product development platform for the emergence of next generation TIS that are capable of exhibiting higher level of intelligence compared to existing TIS. It also facilitates the proactive evolution of such TIS in tandem with emerging user requirements in the tourism domain. Initial experiment through a case study in product line development gave credence to the viability of the notion of PLOSATIS. PLOSATIS is currently an ongoing work, and so in future, industrial case studies of the application and adoption of PLOSATIS will be conducted to further validate it.

References

1. Steinbauer A. (2005): Consumer Behaviour in e-Tourism, Doctoral Thesis, Department for Information Systems and e-tourism Ludwig-Franzens-Universität Innsbruck (2002).
2. Werthner, H. and Klein, S.: Information Technology and Tourism—A Challenging Relationship, Springer-Verlag, New York, 23. (1999)
3. Staab, S., Werthner, H., Ricci, F., Zipf, A., Gretzel, U., Fesenmaier, D.R., Paris, C., and Knoblock, C.: Intelligent systems for tourism, IEEE Intelligent Systems, Volume 17, Issue 6, Nov/Dec, 53-66. (2002)
4. Daramola, O., Adigun, M., Ayo, C., Building an Ontology-based Framework for Tourism Recommendation Services, ENTER 2009, pp. 135-147, Amsterdam, Netherlands (2009).
5. Clements P., Northrop L.: Software Product Lines, Practices and Patterns, The SEI Series in Software engineering, Addison-Wesley. (2002)
6. Maedche, A. and Staab, S. Applying Semantic Web Technologies for Tourism Information System. Proceedings of the 9th International Conference for Information and Communication Technologies in Tourism (ENTER 2002), Innsbruck: 113-124 (2002).
7. Lima, F., and Schwabe, D.: Application modeling for the Semantic Web. In Proceedings of the 1st Latin American Web Congress, Washington, DC, USA, IEEE Computer Society, 93-102, (2003).
8. Schwabe, D., Szundy, G., Silva de Moura, S., and Lima, F.: Design and Implementation of Semantic Web Applications. WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web. New York, USA, May 2004, CEUR-US.org.
9. Jin, Y., Xu, S., Decker, S., and Wiederhold, G.: Managing Web Sites with OntoWebber. In Proceedings of the 8th International Conference on Extending Database Technology, Prague, Czech Republic, Springer, 766-768, (2002).
10. Vdovjak, R., Frasinca, F., Houben, G.J., Barna, P.: Engineering Semantic Web Information Systems in Hera. Journal of Web Engineering (JWE), Vol. 2, Nr. 1-2, Rinton Press, 3-26. (2003)
11. W3C Time, <http://www.w3.org/TR/2006/WD-owl-time-20060927/>, accessed: 12/12/2010
12. W3C Geo, <http://www.w3.org/2003/01/geo/>, accessed: 12/12/2010
13. <http://jena.sourceforge.net/> accessed: 08/10/2009.
14. <http://clarkparsia.com/pellet/> 10/11/2009
15. Kang, K., Cohen, J., Novak, W. and Peterson, S., "Feature-oriented domain analysis feasibility study", Carnegie Mellon University, Software Engineering Institute, *Tech. Rep. CMU/SEI-90-TR-21*, pp. 1-82, (1990).
16. Tyrrell, S.: The Many Dimensions of the Software Process. *ACM Crossroads*. Summer 2000, pp. 22-26, (2000).
17. Basili, V., Caldiera, G., and Rombach, H.D.: Goal Question Metric Approach, Encyclopedia of Software Engineering, pp. 528-532, John Wiley & Sons, Inc. (1994)
18. Bockle, G, Clements, P., McGregor, J.D., Mathig, D., Schmid, K: Calculating ROI for Software Product Lines, IEEE Software, 21(3), pp. 23-31. (2004)
19. Paslaru-Bontas Simperl, E., Tempich, C., and Mochol, M.: Cost estimation for ontology development: applying the ONTOCOM model, In W. Abramowicz and H.C. Mayr, Technologies for Business Information Systems, pp. 327-339. Springer, Netherlands. (2007)
20. Boehm, B., Abts, C., Winsor Brown, A., Chulani, S. Clark, B.K., Horowitz, E., Madachy, R., Reifer, D., Steece, B.: "Software Cost Estimation with COCOMO II", Upper Saddle River, N.J. (2000)
21. Ahmed, F. and Capretz, L. F.: A Framework for Process Assessment of Software Product Line, Journal of Information Technology Theory and Application (JITTA), 7:1, 135-157. (2005)