Standards Supporting Autonomic Computing : CIM
Remodeling "LocalizationCapabilities" class in Core Model

A Thesis
Presented to the Graduate School,
Faculty of Engineering – Alexandria University - Egypt

For the Degree of
Master of Science
In
Computer Science

Field of Specialization
Autonomic Computing

By
Rania Al-Maghraby
(Rania@islamway.net)

June 2007

Abstract

Autonomic Computing is a steadily emerging and promising research field. It aims at simplifying interoperability to diminish the management complexity in several industries. To this goal, some standards have been worked on to establish a common basis for communication and interaction on all management areas. This work focuses on one of these standards, CIM – developed by DMTF, trying to enhance its support for localization handling by proposing some modifications and additions that are intended and designed to facilitate the configuration of different localization aspects.

1. Introduction :

As enterprises strive to meet their current challenges, they require an IT infrastructure that supports their business goals. An IT infrastructure that enables business to be more responsive, variable, focused, and resilient.

Autonomic systems are such systems that are self-configuring, self-healing, self-protecting and self-optimizing. They are "Intelligent" open systems that manage complexity, know themselves, continuously tune themselves, adapt to unpredictable conditions, prevent and recover from failures and provide a safe environment. They let enterprises focus on business, not on IT infrastructure.

For autonomic computing technology, many relevant standards and their associated standards organizations are described in "An Architectural Blueprint for Autonomic Computing" (see References). In our work, we examine selected autonomic standard in detail; CIM.

More specifically, the handling of CIM model is conducted with concentration on the localization support features in the current version of the model (V2.15), by enhancement and additions to it to be more mature in dealing with localization configuration settings.

Software localization is, by global consensus, an extremely important issue in IT industry. It's agreed upon that understanding different cultures and marketing for them is a critical success factor. Generally, Localization means allowing an application/product interface to adapt with specific locales (languages, countries, cultures, …). There are several techniques and tools that support localization.

The LocalizationCapabilities class in CIM core model was introduced in previous versions of CIM (V2.9) with the purpose of supporting diagnostic service data, with the intent that it will be generalized for broader use in future. It was then deprecated upon a change request to replace localization features in the schema by usage of the protocol CIM/XML over HTTP. Refer to more details on criticism of current design and rationale and significance of proposed work in section 6.4.

2. Autonomic Computing Overview :

2.1. Definition: Autonomic computing systems consist of four attributes. As illustrated in the following 4-quadrant chart, they are:
_ Self-configuring (able to adapt to changes in the system)
_ Self-healing (able to recover from detected errors)
_ Self-optimizing (able to improve use of resources)
_ Self-protecting (able to anticipate and cure intrusions)

2.2. Autonomic Computing Concepts:

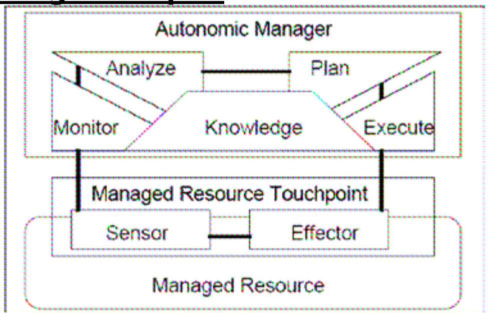


Figure 1. Autonomic Computing Control Loop

2.3. Autonomic Maturity Levels : There are typically five levels of autonomic maturity, which are measures of where any given process is on that spectrum. These maturity levels are: Basic – Managed – Predictive – Adaptive – Autonomic.

2.4. Value of Autonomic Computing : Few examples of the results delivered by implementing autonomic computing solutions with self-management characteristics are : Operational efficiency, Supporting business needs with IT, Workforce productivity. Systems that are self-managing free up IT resources which then can move from mundane system management tasks to focusing on working with users to solve business problems.

3. Standards Overview :

3.1. Standards Organizations : DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

3.2. Key Standards : CIM is a DMTF standard for expressing data about systems, applications, networks, and devices. It is a conceptual information model for describing computing and business entities in enterprise and Internet environments.

3.3. Perception of Autonomic Computing Standards in Marketplace :
Since autonomic IT systems will likely be comprised of multiple components from multiple vendors, it is critical that the various system components be able to communicate with and understand one another. That's why IBM is an active leader in the development of open standards for autonomic computing. This is a significant challenge because of the many different technology areas that autonomic computing spans.

In November 2004, IBM and Fujitsu, a large and influential Japanese IT corporation, announced their agreement to collaborate on autonomic computing standards. This event was part culmination and part catalyst; it was the result of discussions between the two companies about their shared interest in autonomic computing technology and in developing and promoting open standards on which autonomic computing architecture could be based. It also sparked additional interest in autonomic computing technology in Japan, leading to a follow-up announcement about one week later.

This agreement between two very large IT corporations to collaborate on open standards that are critical to autonomic computing technology was widely publicized, but it was not isolated. A press release from Tokyo on December 6, 2004 noted several additional companies that embrace autonomic computing technologies and announced that they would join IBM's autonomic computing initiative.

These companies joined four others, already announced as IBM autonomic computing business partners, who have already deployed autonomic computing capabilities or standards from the Autonomic Computing Toolkit. The list of Japanese companies who have embraced autonomic computing technology reads like a significant portion of a virtual "Who's Who in Computing in Japan" (including, of course, IBM Japan).

Autonomic computing continues to become more widely known, with increasing adoption in the IT industry. IBM has many autonomic computing business partners from around the globe. Even so, there seems to be an especially sharp interest in autonomic computing in Japan. [6]

4. CIM – Common Information Model :

4.1. Overview : The CIM provides object classes, properties, methods, and associations common to the use of management applications in the form of management schema. These are organized into three layers:

- Core model : an information model that captures notions that are applicable to all areas of management. While it is possible that additional classes will be added to the Core model over time, major re-interpretations of the Core model classes are not anticipated.

- Common model : an information model that captures notions that are common to particular management areas, but independent of a particular technology or implementation. The common areas are systems, applications, databases, networks and devices. The information model is specific enough to provide a basis for the development of management applications. This model provides a set of base classes for extension into the area of technology-specific schemas. The Core and Common models together are expressed as the CIM schema.

- Extension schemas : represent technology-specific extensions of the Common model. These schemas are specific to environments, such as operating systems (for example, UNIX or Microsoft Windows).

4.2. Current Incorporation of Localization in CIM models (CIM 2.15) : A LocalizationCapabilities class is defined in the core schema, as a subclass of the Capabilities class, which is a type of ManagedElement class, and represents capabilities defined for (associated with) a certain ManagedElement.

4.3. Work Scope : CIM's goal is to model all the various aspects of the managed environment, not just a single problem space. Coping with that goal, we propose enhancements and additions to the LocalizationCapabilities class in the Core model, in order to allow it to support the guidelines and requirements of software localization design and development techniques in a practical way. The objective is to allow the standard to provide a model for multilingual localizable managed resources.

5. Software Globalization/Localization:

5.1. Definition : Localizing the UI of a particular application involves translating the text in localization packs, modifying images and icons as necessary for regional considerations, and modifying layout to accommodate text or image-size changes.

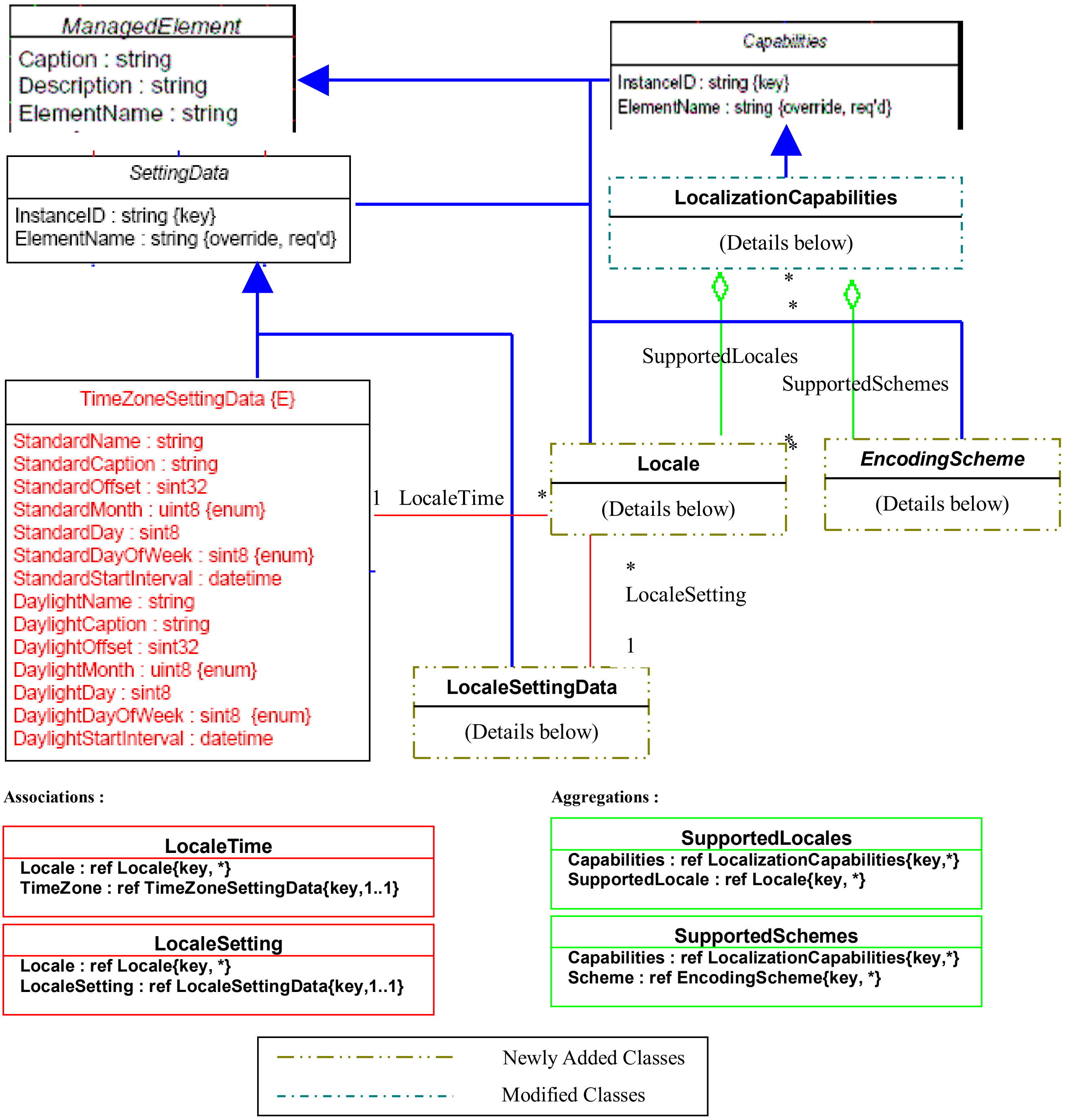
5.2. Locale : Locale is used by the software industry in general to mean any of the following related concepts :
- The set of people who share a set of common expectations about their computer interactions
- The common expectations of computer behaviors that those people share
- The name given to one of those particular sets of expectations or people
- The computer-readable data (and sometimes code) that encapsulates those behaviors

A locale model contains assumptions about all of these cultural features. Frequently used locale models include Numbers – Currency – Date – Time – Calendar - Telephone – Measures – Icon – Conventions – Sorting orders - … .

5.3. CLDR (Common Locale Data Repository) : A Relatively new project started in 2004, hosted by Unicode Consortium. CLDR is by far the largest and most extensive standard repository of locale data.

6. Proposed Model for “LocalizationCapability” Class :

<u>6.1. Model Modifications & 6.2. Proposed Changes :</u> Models are abstractions of "real world" objects and events. In this notion, the proposed model modifications abstract the localization objects and events in an operating environment. Data flow and configuration perspectives are taken in consideration. Changes made preserve the model backwards compatibility.



ManagedElement
Caption : string
Description : string
ElementName : string
Capabilities
InstanceID : string {key}
ElementName : string {override, req'd}
SettingData
InstanceID : string {key}
ElementName : string {override, req'd}
LocalizationCapabilities
(Details below)
*
*
SupportedLocales
SupportedSchemes
TimeZoneSettingData {E}
StandardName : string
StandardCaption : string
StandardOffset : sint32
StandardMonth : uint8 {enum}
StandardDay : sint8
StandardDayOfWeek : sint8 {enum}
StandardStartInterval : datetime
DaylightName : string
DaylightCaption : string
DaylightOffset : sint32
DaylightMonth : uint8 {enum}
DaylightDay : sint8
DaylightDayOfWeek : sint8 {enum}
DaylightStartInterval : datetime
1
LocaleTime
*
*
*
Locale
(Details below)
EncodingScheme
(Details below)
*
LocaleSetting
1
LocaleSettingData
(Details below)
Associations :
LocaleTime
Locale : ref Locale{key, *}
TimeZone : ref TimeZoneSettingData{key,1..1}
LocaleSetting
Locale : ref Locale{key, *}
LocaleSetting : ref LocaleSettingData{key,1..1}
Aggregations :
SupportedLocales
Capabilities : ref LocalizationCapabilities{key,*}
SupportedLocale : ref Locale{key, *}
SupportedSchemes
Capabilities : ref LocalizationCapabilities{key,*}
Scheme : ref EncodingScheme{key, *}
Newly Added Classes
Modified Classes

Figure 2. Proposed Model for “LocalizationCapabilities” Class

6.2. Classes Details :

CIM_LocalizationCapabilities

IsLocalized : boolean;

IsLocaleSupported(locale:string) : boolean;
IsSchemeSupported(scheme:string) : boolean;

CIM_Locale

FlowDirectionsOrder : uint16 {enum, read}; // rtl, ltr, tb-rl, lr-tb, rl-tb
CourtesyTitlesOrder : uint16 {enum, read}; // Ms., Mrs., Mr., Sir, Dr., Prof., …
GreetingsOrder : uint16 {enum, read}; // G.M., GAN, G.N., Welcome, Bye, …
MeasureUnitsOrder : uint16 {enum, read}; // Weight, Length, Freq., Temp., …
PunctuationSymbolOrder : uint16 {enum, read}; // , . ; / \ : “ ‘ …..
VowelsOrder : uint16 {enum, read}; // a, i, u, e, o, aa, ii, uu, ee, oo, stress, consonant
LocaleIdentifier : string;
LocaleName : string;

CIM_LocaleSettingData

Country : string;
Language : string;
Script : string;
bidiSupport : boolean;
AnnotationSupport : boolean;
VowelizationSupport : boolean;
CurrencyName : string;
CurrencySymbol : char16;
DateFormates : string[]; // Holds format strings available in this locale
// Flags corresponding to the entries in the FlowDirectionsOrder enum
FlowDirections : boolean[];
// Entries in this locale corresponding to the entries in enum data in Locale
PunctuationSymbols : char16[];
CourtesyTitles : string[];
Greetings : string[];
MeasureUnits : string[];
Vowels : [Octetstring] uint8[][]; // Encoded in UTF-8 by default
// Map character to its transliteration, array index in the character hash value
CharTransliteration :string[]; // In English script

CIM_EncodingScheme

SchemeStr : string;
MappingTable : [Octetstring]uint8[][];

EncodeChar([IN]charToEncode : char16) : [Octetstring]uint8[];
DecodeChar([IN, Octetstring]codestring : uint8[]) : char16;

6.3. Significance of Modeling Localization :

Choice of the LocalizationCapabilities class enhancement as our work scope originated from realization of the importance and vitality of localization of several systems in a global flat world. Localization/Globalization is an inevitable requirement in all kinds of industry. This doesn't refer only to language difference, but also to different geographical and cultural aspects. Market hunting in any area/country depends in a big deal on the ability to understand these differences in people nature and to communicate with them according to this understanding, in a way that fits their correct perception. That's why business globalization is a must in today's world.

More explanation of importance and significance of localization modeling in CIM model can be deduced from the following excerpt quoted from "CIM Diagnostics Model White Paper" :

"Localization refers to the support of various geographical, political, or cultural region preferences, or locales. A client may be in a different country from the system it is querying and would prefer to be able to communicate with the system using its own locale. There are inherent differences to be reckoned with, such as language, phraseology, currency, and many cultural oddities.

There is no localization support in CIM prior to Version 2.9. Since diagnostics relies on precise reporting of system status and problem data in a user-centric environment, localization is critical. In V2.9, we introduced schema extensions to allow a client to query a diagnostic service for supported locales and to specify the desired locale via a DiagnosticSetting object. The change was written as generically as possible, specifically supporting diagnostics with the intent that it be generalized for broader use in the future.

A new class, CIM_LocalizationCapabilities : CIM_Capabilities was introduced with properties publishing the supported input and output locals. A Locales[] property is added to the DiagnosticSetting class (for passing to the service) and the DiagnosticServiceRecord class (for local identification of the resultant logs)."

Rationale behind class deprecation and applicability of proposed reactivation can be explained by inspection of the stated reasoning in the change request issued in CIM V2.10, as quoted below :

"There was a discussion at the TC regarding the modeling of localization. Preference is to do localization in the protocol (CIM/XML over HTTP) rather than in the schema. This CR deprecates the localization schema to favor usage of the HTTP Accept-Language and Content-Language header fields as documented in DSP0200, Specification for CIM Operations over HTTP v1.2." [Related excerpts from DSP0200 can be found in appendices].

It's clear that this change request put an assumption that the localization capabilities are limited to different languages support, and didn't pay attention to other aspects of localization. In response to this reasoning, we propose the new model changes to accommodate localization capabilities beyond the language and encoding schemes support. Localization is a broad concept and is not limited to supporting different languages (for more solid background, refer to section 5. Software Globalization/ Localization).

In addition, there is another issue with usability, since the original class included two string arrays only, one for input locales and the other for output locales, which isn't useful to the service intended by introduction of this class, and doesn't add valuable information to serve the needs of configuration of localization capabilities of different elements in the model, since these two array attributes could be simply added to the base class (Capabilities or ManagedElement) without affecting the semantics, and separation in a standalone class is a kind of specialization that could be better exploited by adding more attributes and functionality to this class.

By incorporating the proposed changes the support of configuration of localization capabilities is more mature and useful in serving the needs of model elements to deal with localization aspects of each other and with external systems. This goes on the same line with the above quoted statement from "CIM Diagnostics Model White Paper" after initial introduction of the original class, saying "*The change was written as generically as possible, specifically supporting diagnostics with the intent that it be generalized for broader use in the future."*

---

7. Validation of the Proposed Model :

7.1. Validation Methodology : Basically, the process followed by DMTF for commitment on changes and modification of any of its developed standards depends on discussions between working groups and steering committee resulting decisions to accept or reject changes of each standard release. However, this approach can't be used here for an externally proposed work – not internally generated from DMTF groups discussions. That raises the need for a validation method to show the correctness and rationale of the proposed work.

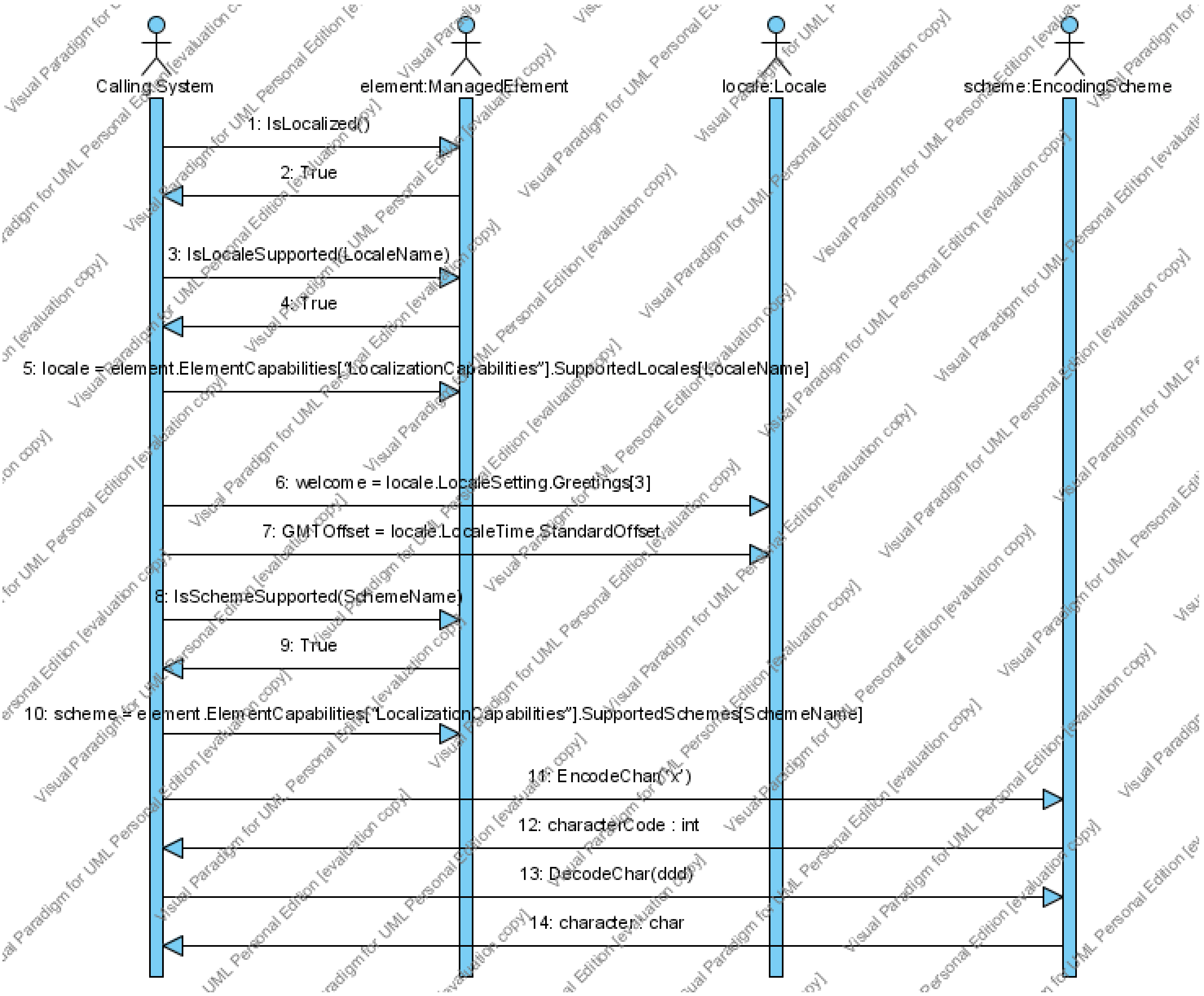
Traditionally, the configuration design of IT service and infrastructure management environments has been a process largely governed by heuristics. In the last few years it became evident that with the business of organizations relying more and more on their IT infrastructure there is a need for the verification and validation of the nonfunctional properties. These service level and information security requirements have to be satisfied in large, heterogeneous IT systems. However, heuristic design of IT management configurations cannot cope with complex systems. Employment of the Model Driven Architecture (MDA) approach, well-established by now, to model, analyze and plan IT infrastructure management configurations can be advantageous. [34]

7.2. How will Standards take advantage of the MDA? Using MDA, the industry gets a standard, every company uses it, none are forced to switch platforms in order to benefit, and the standard lasts as long as the industry process that it serves. Everybody wins! [27]

7.3. An MDA perspective on proposed work : The CIM model can be regarded as a PIM according to MDA concepts. Starting from this point, the correctness of the proposed model can be validated by clarifying correspondence between the proposed model components and the modeled business requirements. This can be shown as detailed below :

Business Requirement	Model Representation
Locale is represented in terms of language-territory (country) pairs	Attributes : Locale.LocaleName LocaleSettingData.Country LocaleSettingData.Language Associations : CIM_SupportedLocale CIM_LocaleSetting
Locale is used as a mechanism to switch language or cultural dependent features, like calendar, time, monetary, formatting, and specific translations of key sentences.	Attributes : LocaleSettingData.Script LocaleSettingData.bidiSupport LocaleSettingData.AnnotationSupport LocaleSettingData.VowelizationSupport LocaleSettingData.CurrencyName LocaleSettingData.CurrencySymbol LocaleSettingData.DateFormates LocaleSettingData.FlowDirections LocaleSettingData.PunctuationSymbols LocaleSettingData.CourtesyTitles LocaleSettingData.Greetings LocaleSettingData.MeasureUnits LocaleSettingData.Vowels LocaleSettingData.CharTransliteration Along with their values & value maps in the Locale class. In addition, all attributes of TimeZoneSettingData class are relevant to this business requirement, they will not be listed here since no changes were proposed for this class. Associations : CIM_LocaleSetting CIM_LocaleTime
Code Page support is one of localization matters. It's necessary for handling characters representation from different scripts and languages.	Attributes : EncodingScheme.SchemeStr EncodingScheme.MappingTable Methods : EncodingScheme.EncodeChar ([IN]charToEncode:char16) EncodingScheme.DecodeChar ([IN,Octetstring]codestring:uint8[]) Associations : CIM_SupportedSchemes

7.4. Sequence Diagram Illustration of Proposed Model Usage Scenarios :


Calling System
element:ManagedElement
locale:Locale
scheme:EncodingScheme
1: IsLocalized()
2: True
3: IsLocaleSupported(LocaleName)
4: True
5: locale = element.ElementCapabilities[“LocalizationCapabilities”].SupportedLocales[LocaleName]
6: welcome = locale.LocaleSetting.Greetings[3]
7: GMTOffset = locale.LocaleTime.StandardOffset
8: IsSchemeSupported(SchemeName)
9: True
10: scheme = element.ElementCapabilities[“LocalizationCapabilities”].SupportedSchemes[SchemeName]
11: EncodeChar(‘x’)
12: characterCode : int
13: DecodeChar(ddd)
14: character : char

7.5. Pseudo Code of implementation prototype :

A Java-like pseudo code of a prototype implementation of the proposed model. It illustrates implementation steps and interaction access methods between the related classes.

The expected output of running the main method of the ExternalSystem class in this prototype is to print the following lines to the standard output :

```
Welcome in AR_EG is : مرحباً
GMT offset in AR_EG is : 2
The code of the first Arabic letter in Unicode is : 0x0623
The character corresponding to code value 0x0669 in Unicode is : ٩
```

8. Conclusion & Future Work :

8.1. Conclusion

In this work, we proposed model modifications to the LocalizationCapabilities class in the core model of CIM standard, which is released by DMTF organization. We clarified the significance of this modification, and set forth validation of model correctness by correspondence to business field requirements. A prototype implementation was presented as well to show the integrity of classes' interactions and achievement of the stated business goal of supporting configuration of localization attributes in the system management domain.

Although implementation of standards is still a little behind in practice, acceptance and adoption of interoperation standards by companies in industry is a roadmap for effectiveness and success of standardization efforts. Experimental industrial applications of proposed standard modifications dictate approval or rejection of the proposed work, as implied by the processing methodology of DMTF organization.

In autonomic computing research, academic and industrial projects realized remarkable parts of the autonomic computing vision. However, there are still open problems to deal with in this promising research area. [28]

This work will be presented to DMTF for further assessment and proposal for inclusion in the organization's standardization efforts in CIM.

8.2. Future Work

As the business requirements and software technologies keep changing and evolving all the way, standards will keep coping with and driving these changes. It's determinate that standards are required to play a more effective role in business development and market place. Continuous investigation and enhancement of the defined standards is inevitable. As new versions of CIM standard come out, revision and revisiting of this work is important. Areas of possible enhancement of the proposed work may include :

- Supporting more localization features; cultural, political, lingual and geographical.
- Modeling the locale data repository (like CLDR) in a more general and manageable manner, like representing it directly in the model by creating specific classes/relations to model it rather than specifying it as a single string field.
- More features of encoding schemes can be exploited, such as character formatting in several format methods.
- Some kind of relation or dependency can be created between the encoding scheme and the locale settings, like coding characters according to known preference in a specific locale. For example, use of specific font or character set in certain locale. This is still achievable using the proposed model but with more concentration on implementation effort, while some future work may leverage this effort to the model itself.
- It may be feasible to add some level of dynamicity to the model to allow locale settings to adapt with the current temporal political and social status.

References

Publications

1- On demand Operating Environment Creating Business Flexibility (Red Book)
2- An Architectural Blueprint for Autonomic Computing [http://www-03.ibm.com/autonomic/pdfs/ACBP2_2004-10-04.pdf]
3- The Autonomic Computing Edge - The Standard way of autonomic computing [http://www-128.ibm.com/developerworks/library/ac-edge2/]
4- Introduction to Autonomic Computing (developer works)
5- A Practical Guide to the IBM Autonomic Computing Toolkit (Red Book)
6- The Autonomic Computing Edge - Autonomic computing heats up in Japan [http://www.ibm.com/developerworks/library/ac-edge/]
7- CIM Concepts White Paper - DSP0110.pdf (DMTF specification)
8- CIM Core Model White Paper - DSP0111.pdf (DMTF specification)
9- Management application programming, Part 2 Introduction to WBEM and the CIM [http://www.ibm.com/developerworks/java/library/j-wbem/]
10-CIM Infrastructure Specification - DSP0004V2[1].3_final.pdf (DMTF specification)
11-CIM_Core.pdf (DMTF specification)
12-e-business Globalization - Solution Design Guide (Red Book)
13-diagnostics_whitepaper_v0_9.51.pdf (DMTF specification)
14-Specification for CIM Operations over HTTP – DSP0200.pdf (DMTF specification)
15-Developing in OMG's Model-Driven Architecture - Jon Siegel and the OMG Staff Strategy Group (OMG White Paper)
16-Ian Sommerville, "Software Engineering", 6th Ed., Addison-Wesley.

Websites

17-http://www.ibm.com/autonomic
18-http://www.dmtf.org
19-http://www.ibm.com/developerworks
20-http://www.ibm.com/redbooks
21-http://www.xencraft.com
22-http://www.i18nguy.com
23-http://www.unicode.org
24-http://www.unicode.org/cldr/
25-http://www.intel.com
26-http://www.lisa.org
27-http://www.omg.org/mda

Academic Papers

28-Autonomic Computing : Emerging Trends and Open Problems - Mazeiar Salehie & Ladan Tahvildari, Dept. of Elect. and Comp. Eng., University of Waterloo. DEAS 2005, May 21, 2005, St. Louis, Missouri, USA.
29-Automating Applications Management in the Enterprise using DMTF Information Models - Umesh Bellur, Indian Institute of Tech. Bombay, Powai, Mumbai
30-Integrating CIM/WBEM with the Java Enterprise Model - Kenneth Carey & Fergus O' Reilly, Adaptive Wireless Systems Group, Department of Electronic Engineering, Cork Institute of Technology, Cork, IRELAND
31-CAMELEON: A CIM "Modelware" platform for distributed integrated management - M. Sibilla, A. Barros de Sales, Y.Steff, T. Desprats, D. Marquié, Institut de Recherche en information de Toulouse (IRIT) & François Jocteur - Monrozier, C. Lasserre, CNES & Anne-Isabe lle Rivière, Alcatel CIT
32-Using CIM to Realize Policy Validation within the Ponder Framework - Leonidas Lymberopoulos, Emil Lupu and Morris Sloman, Imperial College, Department of Computing, 180 Queen's Gate, SW7 2BZ, London, UK

33-A Network Management Platform adaptable to CIM Model evolution - Nathalie Rico, Omar Cherkaoui and Elmi Hassan, Univerisité de Montréal, University of Quebec in Montreal

34-Model-Based Design of System Management - Fault Tolerant Systems Research Group, Department of Measurement and Information Systems, Budapest University of Technology and Economics

35-Impact of Model-Driven Standards - David Flater, National Institute of Standards and Technology, 100 Bureau Drive, Stop 8260 Gaithersburg, MD 20899-8260, U.S.A.

36-"Policy-Maker", a Toolkit for Policy-Based Security Management - Andreas Pilz, Institute for Data Processing, TU München

37-Distributed Network Security - Dipl.-Ing. Oliver Welter, Dipl.-Ing. Andreas Pilz, Technische Universitaet Muenchen