Abstract: The growing complexity of today’s electronic designs requires reusing existing design components, called Intellectual Properties (IPs). The project IPQ (IP Qualification for Efficient Design Reuse) aims to support the engineer by developing reuse-oriented design methodologies and intelligent systems. For this, it is essential to have a proper characterization of IPs as a documentation that can be processed by both, humans and computers. In this paper we present our own developed IP Characterization Language (IPCHL) that is based on current standards for Virtual Component Attributes and actual web technologies like XML. IPCHL enables communication of IP specific knowledge across the World Wide Web and the utilization of knowledge management techniques for flexible and intelligent reuse support.
1 Introduction

The design of electronic circuits is a discipline where two contrasting tendencies can be observed: On the one hand, modern circuit designs get more and more complex and difficult to handle by electronic engineers. On the other hand, global competition requires a continuous reduction of development times. At the same time, the correctness and reliability of the designs should, of course, not suffer from shorter development cycles. These problems have become so dominant that they cannot be met anymore without extensive utilization of design reuse. It is getting vitally important for an electronic engineer to reuse old designs (or parts of them) and not to redesign a new application entirely from scratch. However, the existence and accessibility of reusable designs, which we will further on name Intellectual Property (IP), is only a necessary prerequisite and does not guarantee the reuse of the IP. First, it must be recognized as candidate for reuse and, second, it must be successfully transferred into the development environment of the designer.

In the project “IPQ\(^1\): IP Qualification for Efficient Design Reuse” funded by the German Ministry of Education and Research (BMBF) and the related European Medea project “ToolIP\(^2\): Tools and Methods for IP”, we focus on the development of methodologies and intelligent systems providing assistance to the engineer for all stages of the IP qualification process. Another objective is to enable IP exchange across organizational boundaries and, thereby, enabling virtual marketplaces for IPs to emerge. Our approach presented here leads to the notion of Organizational Memory Information Systems (OMIS), a term originated in the research of techniques and methodologies for knowledge management [Ab98].

By definition, an OMIS is a system that supports a variety of activities like storing, retrieval, or sharing of explicit knowledge in an organization. Here, the term knowledge comprises the IP, a set of documents like VHDL files, manuals, or verification logs.

Unlike traditional databases, data contained in an OMIS is accessed on a semantic layer. In other words, the system has a basic understanding of the information it contains. This allows services on top of the semantic layer to behave smarter when offered to the inside or outside of the organization. For instance, using a technology like CBR [Sc02] for retrieval makes it possible to propose similar IPs to a user in the case that stored IPs do not exactly match his/her requirements. The key factor for semantic based access to the OMIS is a meta-description or characterization of the stored information items, which has to be specified by an engineer in advance.

In this paper we present the XML-based language IPCHL that has been developed specifically for semantic IP characterization. IPCHL adopts several standards like VSIA Virtual Component Attributes (VCA) [VS01b] and OpenMORE Assessment Program for Hard/Soft IP [Sy01]. Furthermore, IPCHL contains constructs for defining taxonomical knowledge and is based on current web technology for semantic annotation in order to achieve interoperability between multiple OMIS. IPCHL is currently evaluated within the IPQ project by several industrial partners and will provide useful input for a future standard language for IP knowledge exchange.

In section 2 we start with a brief introduction of our distributed OMIS scenario. Here, we also present some markup techniques for semantic annotation. Section 3 presents a detailed overview of the different standardization efforts for IP characterization and provides an explanation of IPCHL and an excerpt of the definition. The complete language specification currently under revision will be made available thereafter.

2 Intelligent IP Management by Organizational Memory Information Systems

Before an IP can be reused it must be found and it must be decided about its suitability with respect to a given design situation. If these steps consume too many resources a reuse-oriented design approach will quickly become counterproductive. The basis for a successful reuse management is to store and describe


\(^2\) See http://toolip.fzi.de for partners and further information.
IPs systematically and establish an IT infrastructure providing assistance to designers. Research focusing on these aspects from a conceptual perspective introduced the term *Organizational Memory Information System*. According to [Ab98], an *Organizational Memory Information System* (OMIS) is an intelligent information and assistant system that fosters creation, accumulation, sharing, reuse and further development of explicit knowledge in an organization which may be prevalent in manifold different forms, formats and systems. Explicit knowledge, here, refers to *information items* contained in the OMIS that can be accessed on a semantic layer. Such a layer is made of a proper description of the information that is called *domain ontology* (see [Ba97]). In a broad sense, a domain ontology defines the *meaning* of the information items stored in the OMIS, thereby enabling services like retrieval of IPs and collaboration (e.g. IP transfer) to behave more intelligently. Consequently, for the description of IPs we can distinguish between the following two parts:

1. IP Content: The set of all deliverables related to the IP itself like VHDL files, manuals, or verification logs. These are the information items stored in the OMIS.
2. IP Characterization: A set of elementary properties of the IP Content in conjunction with the domain ontology that structures the properties.

Please note that in our approach IPs are not classified directly by concepts of the ontology. Instead, relevant information is firstly compressed into a vector of properties. This eases the “understanding” of IPs by computers significantly but puts some extra effort on the designer when releasing the IP. However, most of the information carried by the properties is already present in the documentation of an IP and can be extracted from there. Figure 1 shows the scope of the IP characterization by a generic example; the characterization is based on the properties that are associated to particular nodes (or concepts) of the concept graph of the domain ontology. A concept graph represents taxonomical knowledge, e.g. a functional classification of the IP. The IP characterization, that is the combination of IP properties and a domain ontology, can be seen as a *semantic index* or *semantic markup* for the IP content.

![Figure 1: Knowledge Representation for IPs in an OMIS](image_url)

A retrieval engine might choose to answer a question like “Give me all IPs related to concept C (e.g. Multimedia)” by returning concept D (e.g. MP3 Encoder) IPs because it is a sub-concept and related instances might be useful answers, too. An important design principle of an OMIS is to avoid encoding semantics in the application logic of a particular service. Instead, it is explicitly specified by the domain
ontology shared between all services. In our example the retrieval engine is not required to know anything about IPs. The set of services committing to the same ontology is named a community [EA01a] and, in a perfect world, there should be only a single community consisting of all IP providers and IP users.

In practice, companies have their own directives to characterize IPs, perhaps resembling the same techniques as sketched in Figure 1, but with slightly different taxonomies and slightly different properties connected to concepts. For example, one vendor might choose that an MP3 encoder should be classified somewhere below a Multimedia concept comprising the properties frequency and bit rates, while another vendor might have a different conceptualization that is based on other technical considerations like different classes of power consumption or noise immunity. Consequently, the properties used for characterizing IPs may be totally different. Fortunately, there is high evidence that the second vendor also provides information about the frequency and the bit rates of an MP3 encoder somewhere else and an intelligent search tool must extract this information in order to compare the IPs of both vendors. To cope with this, we extend our view toward a distributed OMIS architecture shown in Figure 2 that comprises a society of organizational memory information systems. The challenge is now to develop a commonly agreed language for IP characterization, which we denominated IPCHL. IPCHL contains constructs for defining IP properties and allows tailoring the conceptual knowledge. Beside this, supported properties are not chosen arbitrarily but by adopting current standards like VSIA Virtual Component Attributes (VCA) [VS01b] and OpenMORE Assessment Program for Hard/Soft IP [Sy01]. For a detailed overview of the requirements see [Vi01].

2.1 Web Technologies for Semantic Markup

Since the underlying communication platform for the distributed OMIS architecture is the Internet or, to be more precise, the World Wide Web, IPCHL should not be developed from scratch. Instead, it should
rely on stable and mature Web technologies. Here, the idea of semantic annotation for documents is not new. In fact, it was one of the most popular arguments when the extensible markup language (XML) started to emerge [WC02b]. Nowadays, several XML dialects compete to be the language of choice for semantic markup. We provide a brief overview for reader convenience.

2.1.1 Native XML

XML is a set of rules (you may also think of them as guidelines or conventions) for designing text formats that structure the data. XML makes it easy for a computer to generate data, read data, and ensure that the data structure is unambiguous. Like HTML, XML makes use of tags (words bracketed by '<' and '>') and attributes (of the form name="value"). While HTML specifies what each tag and attribute means, and often how the text between them will look in a browser, XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it. However, it shortly became clear that XML is not the answer to semantic data processing because it is still too low-level. Native XML is used for structured syntax formats while the representation of semantic is left to XML applications like RDF/RDF-Schema [WC02a], XML-Schema [WC02c], or XML Topic Maps [To00].

2.1.2 Resource Description Framework (RDF)

The Resource Description Framework (RDF) on top of XML 1.0 and the RDF specifications provide a lightweight ontology system to support the exchange of knowledge for the Web. RDF is a concept for describing arbitrary kinds of metadata. On the one hand, it defines a data model that is utilized to describe the metadata; on the other hand, it defines syntax to represent the data model in terms of XML. As RDF is based on XML its format is syntactically familiar [WC02a]. The data model makes use of the fact that knowledge can be expressed using (subject, predicate, object) triples. In terms of RDF, such a triple is called a statement. Conceptually speaking, a statement consists of a (subject-) resource, a property (predicate) and the property's value (object), where properties and their values are nothing but special kinds of resources. Alternatively, the values can be literals. At a first glance, RDF is a promising approach. It is powerful enough to express arbitrary kinds of knowledge and even captures fundamental concepts like inheritance. It is simple, based on XML and, regarding its status, already a W3C recommendation. However, there is one essential drawback. RDF does not support primitive data types like numbers or number ranges. Since most properties of IPs will require these data types, it seems more appropriate to choose an XML dialect that directly supports them (especially for automatic validation).

2.1.3 XML Schema

XML Schema is another XML dialect that allows defining schemas for arbitrary kinds of data in XML. XML Schema natively supports primitive data types, while, on the other hand, allows to define your own types which can be atomic or structured (complex). It is important to keep in mind that a schema is only the abstract definition of the data's structure (like a class in OO), whereas the concrete content is specified by an instance of that schema (like an object in OO). An instance of an XML Schema is another XML document whose structure conforms to the schema and is filled with concrete data. The XML Schema - XML instance combination is very powerful. Compared to RDF, it is even more basic and capable of representing arbitrary metadata. As opposed to RDF, it supports primitive data types and cardinalities. Since it is already a W3C standard, there are a lot of tools (like validating parsers) available for processing XML Schema based instance data. In conclusion, we decided to define IPCHL based on XML Schema that, actually, seems the have the greatest maturity of the semantic markup approaches considered so far.
3 The XML-based Language IPCHL

After the introduction of the distributed OMIS architecture and the underlying technical platform we will now focus on the language primitives of IPCHL. We start with an analysis of the current IP standardization efforts that are integrated into the XML Schema definition of IPCHL.

3.1 Standards for IP characterization

For the characterization of IPs two different types constitute the set of potential properties:

- **Application attributes** that refer to properties important to decide about the applicability of an IP in a given design situation
- **Quality criteria** that characterize the IP and its deliverables according to its quality.

Both types are subject to current standardization efforts of the VSIA (Virtual Socket Alliance). For the application attributes, the document „Virtual Component Attributes (VCA) With Formats for Profiling, Selection and Transfer Standard Version 2 (VCT 2.2.x)” [VS01b] presents a variety of attributes together with proposals for their syntactical representation. For the quality criteria, the decision about several standardization proposals donated by the VSIA members is still pending. A candidate, for example, is the OpenMORE Assessment Program from Synopsis [Sy01]. For the characterization of IPs by elementary properties it is essential to be compliant with the standards of the VSIA. It will ensure a wider acceptance because it is expected that tools like design checkers will commit to these standards, too. However, because of the fact that the standardization is still in progress, it is necessary to rely only on the stable parts and allow enhancements as required. In the remainder of this section we will have an in-depth view on the VSIA characterization properties being relevant.

3.1.1 IP Application Attributes

The IP Application Attributes are the most important information about an IP. Let us first look at an example attribute taken from the VSIA VCA document [VS01b] with slight modifications for clarification:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>ip(characterization/IP Application Attributes/VC Provider Claims/Functional Overview/Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>single: M..256 (ISO6093)</td>
</tr>
<tr>
<td></td>
<td>Textual description of the functional behavior of the VC</td>
</tr>
<tr>
<td>Classification:</td>
<td>multiple: M..128 (ISO6093)</td>
</tr>
<tr>
<td></td>
<td>A value from the Functional Class Taxonomy (see text)</td>
</tr>
<tr>
<td></td>
<td>VSIA Attribute Example</td>
</tr>
</tbody>
</table>

Table 1: VSIA Attribute Example

Each attribute has a name associated as shown in the first row of Table 1. The names are structured within a categorization that defines the namespace partially shown in Figure 3. Each leaf of this tree (only the leaves for the category VC Provider Claims are shown) can be a value of the attribute name. Originally, the VSIA VCA standard only defines the nodes below IP Application Attributes in the namespace. Because IPCHL has to consider the IP quality criteria as well, the namespace has been enhanced accordingly. Depending on the value of the attribute name the other two fields are Description with the type “single: M..256” and Classification with the type “multiple: M..128”. The type “single: M..256” defines a single value of mixed characters up to a length of 256. An example is a description like “Interpretive JAVA execution engine with enhanced class library”. The signature ”M..256” is a standard type defined by ISO 6093. Consequently, the Classification field can have multiple values up to a length of 128; each value must be taken from a functional class taxonomy that is referred to as a “pick-list” in the VSIA document. This taxonomy is a large hierarchy structuring the different functionalities an IP can fulfill. To give an impression, an excerpt is shown in Figure 4.
From a knowledge engineering point of view, restricting the values of the field Classification to text strings fulfilling the “M..128” constraint and taken from the taxonomy of Figure 4 is equivalent to considering the taxonomy as the type. Hence, we define the type of this field as a collection of values defined by the functional class taxonomy of the VSIA VCA.

As mentioned before, depending on the value of the Attribute Name field, or for short “the name”, there exist other fields for the attribute description. An example is an attribute with the name “ip/characterization/VC Provider Claims/Performance/Frequency” aggregating the fields Minimum, Typical, and Maximum that have numerical types and a Unit field associated (that is, each field is complex and contains the raw value and the unit).

In the remainder of this document we will consider IP Application Attributes as in the following 

**Clarification** (IP Application Attribut): 

An IP Application Attribute is a tuple \((n,v) \in N \times V\) with \(n\) a hierarchical name from the VSIA compliant namespace \(N\) as shown Figure 3 and \(v\) the value of the attribute with type \(V\). It is \(V=V_1 \cup V_2\) and \(V_1\) is a type aggregating basic values like integer or real and information about the measurement unit. \(V_2\) is a taxonomic type that is specified by a node of an additionally defined taxonomy or by a collection of nodes from the same taxonomy. Neither does this clarification include the representation of intervals like it is required by the VSIA for minimum, typical, and maximum values, nor collections of arbitrary types (only collections of taxonomic types are considered). For the moment, this clarification is sufficient. Later, extensions for formalizing parameterized IPs will be defined.

3.1.2 IP Quality Criteria

Unlike IP application attributes, standardization efforts for IP quality criteria have emerged recently and current proposals are very immature. On the other hand, their structure is less complex than IP application attributes. Hence, the following assumptions about the basic characteristics of IP quality criteria can be made without loss of generality:

1. To each quality criterion a simple data type can be assigned. The type defines the range of possible values for quality rating. For example, a type can define score values from 0 (worst) to 6
(best). The type of a particular criterion is strictly ordered, like simple score values. Here, order means that a higher value is always “better” than a lower value.

2. There exist quality criteria that are not applicable to all types of IPs. For example, depending on the description language, VHDL or Verilog, different coding guidelines may apply. The last point is covered in current approaches e.g. OpenMORE Assessment Program [Sy01] by a value “Not applicable” for each quality criterion. Again, a hierarchical namespace for IP quality criteria extracted from the OpenMORE Assessment Program is partially depicted in Figure 5.

![Figure 5: IP Quality Criteria Namespace](image)

Analogously to IP application attributes, we consider quality criteria as in the following

**Clarification:** (IP Quality Criteria)

An IP Quality Criteria is a tuple \((n, v) \in N \times V\) with \(n\) a hierarchical name from a standardized namespace as shown by Figure 6 and enhanced in Figure 5 and \(v\) the value of the attribute \(V\). \(V\) is assumed to be is a simple strictly ordered type, like positive integers, sufficient for assessing IP quality criteria.

We conclude the analysis of current IP standards by defining the IP characterization as the union of IP application attributes and IP quality criteria with the extended namespace shown in Figure 6. Please note that a single unified namespace is absolutely essential for identifying particular IP properties even across OMIS boundaries. The namespace and the set of properties standardized so far form the basis of IPCHL and the resulting XML Schema specification will be presented in the next section.

### 3.2 XML Schema Specification of IPCHL

IPCHL was developed as part of an overall IPQ Format comprising IP characterization and IP content as well. Figure 6 shows a tree view on the XML Schema that links both together. The content as well as the characterization is represented as a tree according to the namespace defined before. Every node aggregates a set of sub nodes corresponding to the standardized categories of the VSIA Virtual Component Attributes (VCA) [VS01b] document and OpenMORE Assessment Program for Hard/Soft IP [Sy01]. The elementary IP properties themselves are stored in the leaves. Every node and IP property is optional but if it is present it has the unique name assigned that can be extracted when traversing the particular node toward the root node. Thus, an IP will be represented as XML instance code compliant with the definition of Figure 6. An instance is only valid if it can be checked against the schema-definition without errors. As already indicated, the content of an IP containing all deliverables of the design itself (e.g. VHDL-files, layouts, test-scripts, manuals, data sheets) is connected to the IP characterization on an instance base.

---

\[\text{We omit the illustration of the complete XML Schema tree, here. Interested readers should contact the authors.}\]
The second part of IPCHL addresses the specification of taxonomies. IP taxonomies are tree-like classification schemes expected to be specific for IP providers. It is therefore necessary to allow the definition of arbitrary taxonomies in IPCHL instead of hard-coding them into the specification. Each IP has to be assigned to at least one concept node within the taxonomy. Every node in turn contains links to a set of properties being relevant for all IPs classified by that node. A node „video“, for example, would typically have attributes like „frames per second“ or „frequency“, since these attributes make sense and should be available for all IPs that are classified as „video“ IPs. Nodes inherit the properties from super-nodes. For representation of taxonomical knowledge, IPCHL includes language constructs to define taxonomy trees. These language constructs allow building up a tree that holds some items in every node. Besides the name of the node, a list of links to nodes in the IP properties is stored in every node. Furthermore, each node can include child-nodes in order to build up the tree structure. These items allow the association of an arbitrary set of IP characterization properties with the taxonomy node. The XML Schema implementation is shown in Figure 8, while Figure 9 shows an example. The taxonomy itself is an instance of the corresponding XML Schema of Figure 8. With this formal XML Schema description it can be ensured that all instances based on this mechanism for defining taxonomies are valid taxonomies of the IPCHL.

3.3 Future IPCHL Extensions

In this section we give a brief outlook at extensions of IPCHL that will be a future research topics in the IPQ Project. We start with parameterized IPs playing an important role in the development of IPs as reusable components. Furthermore, a certain degree of parameterization is always present for mixed-signal IPs.

3.3.1 Parameterized IPs

The basic motivation for parameterized IPs is the greater flexibility that may lead to a higher reuse factor of an IP. On the other side, the development effort for a parameterized IPs is definitely higher and it is currently not clear, what kind of parameterization justifies these additional expenses. However, IPCHL
should be aware of parameterized IPs and, for that reason, we already have proposed general requirements for parametrized IP representations in [Vi01].

Figure 7: Excerpt of the IPQ Transfer Format (XML Schema representation)

A parameterized IP is an IP that allows the customization in some way. This can be established by offering ranges instead of fixed values for some of the properties of the IP characterization or by allowing additional parameters that must be specified by the user when customizing the IP.

A simple example for a parameterized IP is a description of a multiplier where the word length is a parameter that can be varied between 4 and 32. Concrete values for such parameters must be determined before the design can be synthesized (soft IP) or when it is included in the current design (hard IP). A representation language is required to support the explicit specification of dependencies by constraints.

Figure 8: IPCHL constructs for defining taxonomies
Because IPCHL is based on XML Schema basic data types with a well known semantic like integer or real values are already available and the formalization of constraints will be made by mathematical expressions. In general, arbitrary mathematical functions are expected to be too complex because constraints must be computable in both directions.

<?xml version="1.0" encoding="UTF-8"?>
<taxonomy xmlns="http://www.upb.de/cs/ipl/ipq" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.upb.de/cs/ipl/ipq taxonomy.xsd">
  <node name="Physical Library">
    <link>LinkToAttribute1</link>
  </node>
  <node name="Memory Element">
    <link>LinkToAttribute1</link>
    <link>LinkToAttribute2</link>
    <node name="RAM">
      <link>LinkToAttribute3</link>
      <link>LinkToAttribute4</link>
    </node>
    <node name="Cache">
      <link>LinkToAttribute3</link>
      <link>LinkToAttribute5</link>
    </node>
    <node name="Video RAM">
      <link>LinkToAttribute4</link>
      <link>LinkToAttribute6</link>
    </node>
    <node name="Static RAM">
      <link>LinkToAttribute7</link>
      <link>LinkToAttribute8</link>
    </node>
  </node>
  <node name="Non-Volatile">
    <link>LinkToAttribute9</link>
  </node>
  <node name="Analog and Mixed Signal">
    <link>LinkToAttribute4</link>
  </node>
</taxonomy>

Figure 9: Example IPCHL taxonomy

4 Conclusion

The design of electronic circuits increasingly requires the incorporation of design reuse. To realize reuse of existing designs it is necessary to find, evaluate and transfer IPs. To support these processes, we described a distributed OMIS scenario in chapter 2.

Regarding this scenario it is essential to constitute a language for knowledge exchange that incorporates standards as much as possible by still being flexible enough to meet the real world assumptions. Driven by the OMIS scenario we presented the language IPCHL knowledge based characterization. To find an appropriate base for IPCHL we discussed different web technologies for semantic markup. XML Schema seems to be an appropriate base for the technical realization because it has the greatest maturity of the semantic markup approaches considered until now.

The standards for IP characterization from the VSI Alliance and the OpenMORE Assessment Program are treated in chapter 3. These standards are the basis for the XML Schema specification of IPCHL. Besides the unambiguous and for all IP users and IP providers equal attributes mentioned in these standards the IP taxonomies are an important part of the IPCHL definition because they are tree-like classification schemes expected to be specific for IP providers.

Furthermore, in chapter 4 we sketched some future extensions to IPCHL for parameterized IPs. Other extensions for the future will address the specification of so-called similarity models, which reflect essential ontological knowledge for a CBR-based retrieval mechanism.
References


[WC02a] W3 Consortium, Resource Description Framework (RDF), http://www.w3.org/RDF