International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.4, July-Aug. 2012 pp-2568-2573 ISSN: 2249-6645

Consistency Evolution of Process models based on Structural Analysis and Behavioral Profiles

CH. Gurunatham[1], P. Rajarajeswari[2], Dr. D. Vasumathi[3]
* (M.Tech, Computer Science and Engineering, Madanapalle Institute of Technology & Sciences, Madanapalle, A.P,India)
** (M.Tech[Ph.d], Assistant Professor, Department of CSE, Madanapalle Institute of Technology & Sciences, Madanapalle, A.P, India)
*** (M.Tech, Ph.d, Associate Professor, Department of CSE, JNTU Hyderabad, Hyderabad, A.P,India)

Abstract: *To turn the business application of a software project to system specification is a big challenge in business environment. Since the business analysts and system analysts have their own perspective, the modeling of business processes is necessary to facilitate both the perspectives and for a better coordination. Many applications of such business processing model have given raise to problems at support models. As such maintaining consistency at such related models has become a big challenge for business modeling theory and practice. In this paper we be projected a theory to find out the consistency between the models is not coined and it is very much necessary to find out the inconsistencies so that change propagation between the models can be achieved. Behavioral profile, as a solution to the inappropriateness of behavioral equivalence notions will be proposed. This model is expected to resolve the behavioral constraints of a process model. Through this model, profiles can be computed efficiently in public time for sound free-choice Petri nets with reference to their number of places and transitions. We develop a slow model that verifies process consistency described in a Petri net graph. We provide both Structural Analysis and Behavioral Profiles to evaluate process consistency.*

Key Words: *Process Models, Petri nets, Behavioral Profiles, Structural Analysis, Process Modeling.*

I. Introduction

Translating business requirements into system specification is a crucial task of any software engineering project in a business environment. The modeling of business processes has been identified as an important step towards bridging the gap between business and software development, and, among others, facilitating structured design [1], business-IT alignment [2], or engineering of process-aware information systems [3]. There are different solutions that should contribute to a smooth progression from business analysis to software implementation. Methodologies for integrated system design propose to derive technical realizations from business requirements directly via refinements [4], [5], [6]. In the same vein, the standardization of the Business Process Modeling Notation (BPMN) [7] by the OMG received much attention, due to the translation to the Web Services Business Process Execution Language (BPEL) [8] that is part of the specification. There are also various tools on the market that support business process modeling and corresponding transformations. We already mentioned that business analysts and software designers tend to model the same business process in quite different ways, which often impedes efficient communication. Clearly, pragmatics is an inherent feature of every conceptual model. While mapping and reducing the reality are essential for creating model, the purpose of the model determines what to map and what to reduce [11]. As business analysts and software designers have quite diverging concerns when looking at a business process, it is no surprise that business process models differ from software design models of the same process significantly. We will argue throughout this article that a formal concept for discussing the consistency of an alignment between two process models is missing. It is needed for identifying inconsistencies, as well as to enable change propagation between these models. In the software engineering community, consistency refers to a ‘degree of uniformity, standardization, and freedom of contradictions’ [12].

An alignment of process models requires the identification of model correspondences, which is a well-researched topic in the database community. Correspondences relate elements that have matching semantics in the context of an alignment of two models (note that the semantics might differ in absolute terms). Given a set of correspondences, the question whether two data models are consistent is similar to the question whether a mapping between data schemas is valid, which is known from the field of data integration. In this area, various properties for evaluating the validity of a schema mapping have been proposed. For instance, compatibility of a mapping between two schemas requires the existence of a pair of instances that satisfies the constraints of the mapping as well of the respective schemas [13]. Translated into the domain of behavioral models, this yields a consistency notion, which requires the existence of a single trace that is possible in both models after the corresponding elements have been resolved. Again, we might draw the analogy to behavioral models under the assumption of behavioral constraints as the elementary elements. For a given projection between two models, that is a partial correspondence relation, all behavioral constraints on traces of one model are preserved in the traces of the other model. There is a multitude of equivalence criteria in the linear time – branching time spectrum [14]. This criterion is still rather strict, and might not be appropriate for deciding on a consistent alignment between two process models. First, trace equivalence is not invariant to so-called forgetful refinements of activities [15].Forgetful refinement refers to a change, in which an activity is forgotten due to its replacement with an empty activity. As a consequence, we argue that, for our purpose, a notion of consistency that guarantees' freedom of contradictions' as required by [12] should be less strict than a notion requiring all information of one model to be present in another model as well. Furthermore, all behavioral equivalence criteria being discussed in this area, including trace equivalence [14], provide a true/false result.

We introduce the formal concept of a behavioral profile and structural analysis. These profiles capture the essential behavioral constraints of a process model and apply the structural analysis, such as mutual exclusion of activities or partial order. The behavioral profile enables us to overcome three major shortcomings of an application of trace equivalence in an alignment scenario and structural analysis verifies the accuracy and consistency measures captured during behavioral profiling.

1) Behavioral profiles are less sensitive to projections than trace equivalence. We will show that behavioral profiles of two process models remain unchanged even if additional start and end branches are introduced in one of the models.
2) The structure of a behavioral profile provides us with a straight-forward way to define a degree of consistency ranging from 0 to 1.0, referred to as the degree of profile consistency. In this way, we can feed back detailed information to business analysts and software designers on how far and where two models deviate from each other.
3) The concept of a behavioral profile builds on formal properties of free-choice Petri nets. This class of nets has been used for the formalization of most process modeling languages. The derivation of a behavioral profile and the calculation of a degree of profile consistency and structural analysis of the consistency measured have been implemented to demonstrate the applicability of our approach. In this article, we also report the findings from checking consistency between partially overlapping of example process models, a collection of benchmark process models that describe the functionality of specific business software. We introduce consistency measurement by using behavioral profiles and structural analysis in section2, We provide consistency measurement for aligned process models in section3, and we also provide Experiment and Result analysis in section4.

II. Consistency Measurement by using Behavioral Profiles and Structural Analysis

Business process change is at the very core of business process management, which aims at enabling flexible adaptation to changing business needs. However, the wide variety of drivers for business process modeling initiatives, reaching from business evolution to process enactment, results in multiple models that overlap in content due to serving different purposes. That, in turn, imposes serious challenges for the propagation of changes between these process models.

Now a day, Business Process Management (BPM) has a broad field of application, reaching from process evolution to process enactment. The *purpose* guides the creation of every particular process model. It is a consequence of this observation that companies create different models for the same process. These models reside on different levels of abstraction and assume different modeling perspectives depending on what is appropriate with respect to the modeling goal. The flexibility to adapt business processes in order to respond to changing business needs is at the very heart of BPM. Therefore, the *propagation of changes* between several related process models is a major use case for model alignment. According to Gartner, change is of high relevance to the key elements of the BPM discipline, which are *'keeping the business process model in sync with process execution [and] enabling rapid iteration of processes and underlying systems for continuous process improvement and optimization'* .

In our proposal presents a novel approach to change propagation between business process models. Its central contribution is the definition and application of a technique for dealing with overlapping process models that are not defined in terms of a hierarchical refinement. This technique is based on the notion of a behavioral profile which captures a set of dedicated behavioral aspects of a process model. Given a change in the source model, our approach isolates a potential *change region* in the target model grounded on the behavioral profile of corresponding activities. In this way, process modelers can quickly assess the necessity to propagate the change. If change propagation seems to be appropriate, the change region spots the position where to extend the model.

A. Process Models

Our notion of a process model is based on a graph containing activity nodes and control nodes, which, in turn, captures the commonalities of process description languages.

Behavioral Profiles

The Behavioral profile aims at capturing Behavioral aspects of a process in a fine-grained manner. That is, it consists of three relations between nodes of a process graph. These relations are based on the notion of weak order. Two nodes or flow arcs of a process model are in weak order if there a trace in which one node occurs after the other. Note that we require only the existence of such a trace. Thus, weak order does not have to hold for all traces of the model.

○ The strict order relation $x \succ P(y) \text{ and } y \neg\succ P(x)$

○ The exclusiveness relation $x \neg\succ P(y) \text{ and } y \neg\succ P(x)$

○ The observation concurrency relation $x \succ P(y) \text{ and / or } y \succ P(x)$

The set of all three relations is the Behavioral profile.

Two process models with equivalent behavioral profiles may differ in the trace equivalence, in contrast the two process models with identical trace equivalence can also identical in behavioral profiles.

Correspondence Relation: if the relation between two process models is left unique and is not functional

Aligned Transitions: let a1, a2 correspondence to a and c1, c2 correspondence to c. if transition observed from a1 to c1 ,a1 to c2, a2 to c1 or a2 to c2 then the transition relation between a to c is aligned transition.

Projected Firing sequence: In a sequence considered, the set of aligned sequences is referred as firing sequence.

Trace Consistency of Alignment: If Aligned transitions of a projected firing sequence contain trace equivalence then it reflects as Trace consistency of alignment.

B. Structural analysis: The structural analysis of dynamic lumped process models forms an important step in the model building procedure and it is used for the determination of the solvability properties of the model, too. This analysis includes the determination of the degree of freedom, structural solvability, differential index and the dynamic degrees of freedom. As a result of the analysis, the decomposition of the model is obtained and the calculation path can be determined. This way the appropriate numerical method for solving the model can be chosen efficiently. Moreover, advice on how to improve the computational properties of the model by modifying its form or its specification can also be given.

Effective graph-theoretical methods have been proposed in the literature based on the analysis tools developed by, for the determination of the most important solvability property of lumped dynamic models: the differential index. The properties of the dynamic representation graph of process models described by semi-explicit DAE-systems have also been analyzed there in case of index 1 and higher index models. Beside the algorithm of determining the differential index by using the representation graph, a model modification method has also been proposed in the literature, which results in a structurally solvable model even in the case of higher index models.

C. Structural solvability

As a first step, we consider a system of linear or nonlinear algebraic equations in its so called *standard form* where $x_j(j=1,...,N)$ and u_k (k = 1,…,K) are unknowns, $y_i(i=1,...,K)$ are known parameters, $f_i(i=1,...,M)$ and $g_k(k=1,...,K)$ are assumed to be sufficiently smooth real-valued functions. The system of equations above is structurally solvable, if the Jacobian matrix J(x\ u) referring to the above model is non-singular.

$$y_i = f_i(x,u), i=1,...,M \qquad u_k = g_k(x,u), k=1,...,K$$

Consider a system of equations in standard form. We construct a directed graph to represent the structure of the set of equations in the following way. The vertex-set corresponding to unknowns and parameters is partitioned as $X \cup U \cup Y,$ where $X=\{x_1,...,x_N\}, U=\{u_1,...,u_k\}$ and $Y=\{y_1,...,y_M\}$. The functional dependence described by an equation is expressed by arcs coming into y_i or u_k respectively from those x_j and u_I, which appear on its right-hand side. This graph is called the representation graph of the system of equations.

A Menger-type linking from X to Y is a set of pair- wise vertex-disjoint directed paths from a vertex in X to a vertex in Y. The size of a linking is the number of directed paths from X to Y contained in the linking. In case $|X|=|Y|,(M=N),$ a linking of size X is called a complete linking. The graphical condition of the structural solvability is then the following:

Linkage theorem: Assume that the non-vanishing elements of partial derivatives / and graphs, in the standard form model are algebraically independent over the rational number field O. Then the model is structurally solvable if and only if there exists a Menger–type complete linking from X to Y on the representation graph.

We can adapt the graphical techniques to DAE- systems, as well. An ordinary differential equation of a DAE-system can be described by the following equation:

$$x' = f(x_1,...,x_n)$$

Here ***x*** denotes an arbitrary variable depending on time. ***x'*** denotes the derivative of ***x*** with respect to time and $x_1,...,x_n$ are those variables which have effect oil variable ***x'*** according to the differential equation.

In DAE-systems there are two types of variables. ***Differential variables*** are the variables with their time derivative present in the model. Variables, which do not have their time derivative present, are called ***algebraic variables.*** The derivative ***x'*** is called ***derivative*** (velocity) ***variable.***

D. Dynamic representation graph

The value of differential variables is usually computed by using a numerical integration method. Therefore a system of equations including also differential equations can be represented by a ***dynamic graph.*** A dynamic graph is a sequence of static graphs corresponding to each time step of the integration. On a dynamic graph there are directed arcs attached from the previous static graph to the succeeding static graph that are determined by the method applied for solving the ordinary differential equations. In case of a single step explicit method, the value of a differential variable at time ***t+h*** is computed using the corresponding differential value and its value at a previous tune ***t.*** For example, when the explicit Euler method is used:

$$x(t+h) = x(t) + h.x'(t)$$

where ***h*** denotes the step length during the numerical integration. The structure of a dynamic graph assuming explicit Euler method for solving differential equations is shown in ***Fig. 1.***

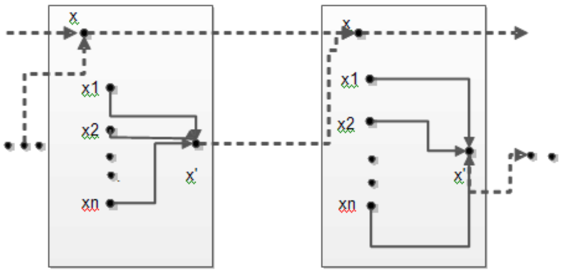

Fig 1: Dynamic Graph model for Euler method

The structural analysis based on graph theoretical technique is carried out in steps performed sequentially. The first step is to rewrite the model into its standard form. The second step is the assignment of types to vertices in the representation graph. The important types of vertices determined by the model specification are the following:

. ***<S>(set)-type variables:*** These represent variables, which are assigned to the specified given values. In the case of a dynamic representation graph assuming explicit method for solving the differential equations, the differential variables will be labeled by type ***<S*>*** because their initial value can be obtained from the initial values, and then their values can be calculated step by step by numerical integration. Labels ***<S>*** and ***<S*>*** are treated the same way during the analysis.

- ***<G>(given)-type variables***: A variable assigned to a specific value of a left hand side is a <G>-type variable. Unlike the <S>-type variables, the values of the right hand side variables will be suitably adjusted so as to preserve the equality of the two sides.

According to the representation graph, the value of every variable which has incoming arcs only from vertices labeled by type ***<S>*** can be calculated by simple substitution into the corresponding equation. These variables become ***secondarily labeled by type <S>,*** and this process can be repeated if necessary. Omitting all vertices labeled primarily, secondarily, etc. by type ***<S>*** and all arcs starting from them from the representation graph we obtain the ***reduced graph.*** The classification of vertices of a reduced graph is as follows:

- all initial vertices form the unknown variable set X
- all terminal vertices labeled by type ***<G>*** constitute the known variable (parameter) set Y,
- all other vertices constitute the known variable set ***Y.***

Dynamic process models can be described by ***semi- explicit DAEs*** as follows:

$$z_1 = f(z_1, z_2, t), z_1(t_0) = z_{I0}$$

(1)

(2) $$0= g(z_1, z_2, t)$$

The most important structural computational property of DAE models is the differential index. By definition the differential index of the semi-explicit DAE (Equations (l)-(2)) is one if one differentiation is sufficient to express z_2 as a continuous function of $z_{1,}$ z_2 and ***t.*** One differentiation is sufficient if and only if the Jacobian matrix g_{z_2} is non-singular.

In our earlier work we have proved that the differential index of the models investigated in is equal to 1 if and only if there exists a Menger-type complete linking on the reduced graph at any time step ***t.***

If the differential index of the investigated model is greater than 1 then there is no Menger-type complete linking on the static graph at any time step ***t.*** The properties of a static graph of a dynamic model, which has differential indexes are as follows.

1. Let us form the following variable sets.

I_0 is the set of the differential variables belonging to the over specified sub graph,

D_0 is the set of the derivative variables referring to the differential variables of set I_0,

I_1 is the set of differential variables from which directed paths lead to the derivative variables in the set D_0,

D_1 is the set of derivative variables referring to the differential variables of set I_1, ... ,

I_k is the set of differential variables from which directed paths lead to the derivative variables in the set D_{k-1},

D_k is the set of derivative variables referring to the differential variables of set I_k ...,

2. Let n be the smallest natural number for which the set D,, contains some derivative variables of the underspecified sub graph. Then the differential index of the model is

$$v_d = n + 2$$

If there is no such number n then the model is not structurally solvable.

In our earlier work we have shown that the important properties of the representation graph including the differential index of the models are independent of the assumption whether a single-step, explicit or implicit numerical method is used for the solution of the differential equations.

III. Consistency measurement for aligned process models:

The previously defined concept of a behavioral profile allows us to formally discuss the notion of a degree of profile consistency between a pair of process models. We will use the classical notion of trace equivalence, which we extend to trace consistency, as a benchmark.

Consistency based on Trace Equivalence

As a benchmark for our consistency analysis, we define a notion of consistency based on the trace equivalence criterion. First, we adapt the trace equivalence criterion for model alignments yielding the notion of trace consistency. Second, the degree of trace consistency is introduced based on the amount of traces of one model that have a counterpart in the other model. We already mentioned in Section 2 that the application of trace equivalence in an alignment setting requires that all parts that have been subject to projection are discarded.

Consistency based on Behavioral Profiles

In general, our notion of consistency based on behavioral profiles, i.e., profile consistency, is grounded on the preservation of behavioral relations for corresponding activities. In contrast to the notion of a trace consistent alignment, it does not require the correspondence relation to be injective. Instead, it allows for 1:n (and even n:m)correspondences. Therefore, this notion can be applied to vertical as well as horizontal alignments. Preservation of the behavioral relation is only required in case there are no overlapping correspondences.

Interpretation of Profile Consistency

As exemplified in the previous section, the degree of profile consistency ranges between 0 and 1.0 for two process models and a correspondence relation. Still, a degree of1.0 does not imply that both models are (projected) trace equivalent. This stems from the fact that the underlying behavioral profile represents a behavioral abstraction; apparently, the degree of profile consistency quantifies the quality of an alignment with respect to the **order of potential activity occurrences. A degree of 1.0** guarantee the all these constraints are equal for the aligned activities of two models. A degree of 0.9, in turn, indicates that the constraints on the order of potential activity occurrences are equal solely for 90% of the relations between aligned activities. However, we assume these thresholds to be highly dependent on a specific project setting. Once a degree of profile consistency below 1.0 is observed, the question of how to locate the source of inconsistency has to be addressed. According to our approach, inconsistencies manifest themselves in different relations of the behavioral profile of two process models for a pair of aligned activities.

This information can directly be provided to business analysts and system analysts in order to judge on the necessity of the inconsistency. While this kind of feedback allows for locating the inconsistency directly in case of only a few inconsistent profile relations (e.g., caused by an interchanged order of two activities in a sequence), it might be inappropriate if a big number of profile relations is inconsistent. Imagine two process models containing a set of aligned activities in sequential order and assume that one of these activities in one model would now be moved to a branch that is executed concurrently to the remaining activities. Then, all behavioral relations between this activity and the remaining activities would be inconsistent, such that feedback on the set of activities that show inconsistent relations would be of little help. Instead, we would consider the biggest subset of aligned activities that show consistent behavioral relations among each other to be valuable feedback on the observed inconsistencies. For the aforementioned case, the single activity having inconsistent relations with all other activities might be identified by this approach.

IV. Experiments and Results Analysis

After preprocessing of the benchmark models, we are able to analyze its consistency. As mentioned before, we establish correspondences between events and functions with equal labels. Further on, we extract all pairs of process models that are aligned by at least two correspondences. For such a pair, we then calculate the consistency measures, that is, trace consistency, the degree of trace consistency, and the degree of profile consistency of the alignment and finally analyzed the accuracy of the degree of profile using structural analysis.

The results are optimistic from the experiments conducted on bench mark business models represented in Petrinet format. We consider the consistency measurement systems WF systems (WF), and Behavior profiling (BP) analysis to compare with the proposed Behavior Profiling and Structural Analysis (BP&SA). We can find the significant benefit of BP&SA over other models. Fig 2 represents the comparison of optimality in consistency measurement between BP&SA and other two models . In fig 3 we can observe the computational over head of the WF. Here BP is having slight advantage over BP&SA, which can be negligible while considering the accuracy achieved through BP&SA in consistency measurement.

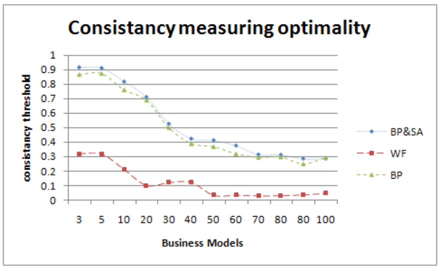

Fig 2: Optimality in Consistency Measurement

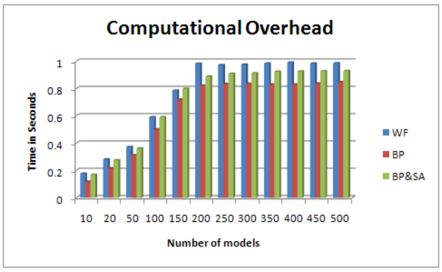

Fig 3: Computational Overhead comparison report

V. Conclusion:

.we have discussed alignment issues between related process models at different abstract levels and different perspectives.. We proposed the concept of behavioral profile that captures the essential behavioral constraints of a process model. Such behavioral profiles are used for the definition of the formal notion of profile consistency. Behavioral profiles provide three major advantages in contrast to the existing notion of trace equivalence and consistency measures that build up it. Finally, the concept of a behavioral profile builds informal properties of free-choice Petri nets. We proved that profile consistency can be checked for sound free-choice WF-systems in O(n3) time with n nodes.

. There are several directions for future research based on behavioral profiles. We have emphasized the fact that different interrelated process models and variants are utilized for the development of process-aware information systems.. We are optimistic that algorithms can be defined to synthesize process model from a behavioral profile, as there exist synthesis techniques to build Petri nets from transition systems and from traces . Such algorithms might not only take one profile as input. We are currently experimenting with building integrated process models from two behavioral profiles and their alignment.

REFERENCES

[1] E. Yourdon, Modern structured analysis. Yourdon Press Upper Saddle River, NJ, USA, 1989.

[2] J. Luftman, R. Papp, and T. Brier, “Enablers and inhibitors of business-IT alignment,” Communications of the AIS, vol. 1, no. 3,1999.

[3] E. Kindler, “Model-based software engineering and process-aware information systems,” LNCS Transactions on Petri Nets and Other Models of Concurrency, vol. 2, pp. 27–45, 2009.

[4] M. Henkel, J. Zdravkovic, and P. Johannes son, “Service-based processes: Design for business and technology,” in ICSOC, M. Aiello. Aoyama, F. Curb era, and M. P. Papazoglou, Eds. ACM, 2004,pp. 21–29.

[5] B. Anderson, M. Bergholtz, A. Edirisuriya, T. Ilayperuma, and. Johansson, “A Declarative Foundation of Process Models,” in CAiSE, ser. Lecture Notes in Computer Science, O. Pastor and J. F.e Cunha, Eds., vol. 3520. Springer, 2005, pp. 233–247.

[6] J. Koehler, R. Hauser, J. M. K¨ uster, K. Ryndina, J. Vanhatalo, and M. Wahler, “The role of visual modeling and model transformations in business-driven development,” Electr. Notes Theor. Comput.Sci. vol. 211, pp. 5–15, 2008.

[7] OMG, Business Process Modeling Notation (BPMN) 1.2, January 2009.

[8] Alexandre Alves et al., “Web Services Business Process Execution Language Version 2.0,” OASIS, Tech. Rep., January 2007.

[9] V. Grover, K. Fiedler, and J. Teng, “Exploring the Success of Information Technology Enabled Business Process Reengineering,”IEEE Transactions on Engineering Management, vol. 41, no. 3, pp.276–284, August 1994.

[10] C. Rolland and N. Prakash, “Bridging the Gap Between OrganisationalNeeds and ERP Functionality,” Requirements Engineering, vol. 5, no. 3, pp. 180–193, October 2000.

[11] T. K¨ uhne, “Matters of (Meta-)Modeling,” Software and System Modeling, vol. 5, no. 4, pp. 369–385, 2006.

[12] IEEE, “Standard Glossary of Software Engineering Terminology,”1990.

CH. Gurunatham, received the B.Tech from JNTU-Anantapur and pursuing M.Tech from Madanapalle Institute of Technology & Sciences (JNTU-Anantapur). He is doing his academic project under the Guidance of Mrs. P. Rajarajeswari.
His Research interests on Software Engineering, Software architecture and Web Technologies.

P. Rajarajeswari, received the B. Tech. from S.V University, Sri Venkateswara University, Tirupathi, Andhra Pradesh.M. Tech degree in Computer Science from JNT University, Hyderabad and Pursuing Ph.D in Computer Science and Engineering, JNT University Hyderabad Andhra Pradesh, India. Presently She is working as Assistant Professor of Computer Science and Engineering in Madanapalle Institute of Technologyand Science, Madanapalle, Chitoor (Dt),Andrapradesh.

Dr. D.Vasumathi, received the B. Tech. from JNT University, Hyderabad, M. Tech degree in Computer Science and Engineering from JNT University, Hyderabad and Ph. D in Computer Science and Engineering JNT University Hyderabad Andhra Pradesh, India. She worked as Assistant Professor, Presently working as Associate Professor in JNT University, Hyderabad .She has 12 years of Teaching experience. Currently guiding 8 Ph. D scholars. She is life member of ISTE and IE. Her research interests Data Mining, Image processing, Web technologies, She has 5 international publications and 10 international conference Publications at International and National level.