$C(A,B)= \alpha 1 * L(A,B)+ 2 * Sem(A,B) + \alpha 3 * Type(A,B) + \alpha 4 * Struct(A,B)$ (5)

The weights (α1,2,3,4) are chosen to be step functions with value = 1.0 until a lower threshold is reached. Thus $\alpha 1=0.0$ for $L(A,B) < 0.9$ (since lexical similarity is a good indication of relationship for only high-scoring matches), and $\alpha 2 = 0.0$ for $Sem(A,B) < 0.67$, $\alpha 3 = 0.0$ for $Type(A,B) < 0.5$, $\alpha 4 = 0.0$ for $Struct(A,B) < 0.5$. The thresholds were derived by computing the similarity per cue for the actual mapping indicated by programmers for integration of candidates services used for testing and taking their average value.

3. RESULTS

We tested the performance of semantic schema matching on 240 distinct pairs of web services drawn from Crossworlds business object library. The business objects tend to have a larger number of member attributes (over 100), so that the algorithm performance could be gauged on large schemas. We then measured the performance by comparing to a manual match of the attributes of the respective schemas. The number of spurious (false positives) as well as missing matches (false negative), were noted in each pair-wise match.

Table 1 illustrates the matching similarly, for a pair of ADTs depicted in Figure 1. Here a web service that provides a description of an inventory item is chained with a web service that retrieves vendor information associated with the inventory item. A match of InventoryType and StockType has been aided by semantic name matching, while abbreviation expansion has allowed match of InvLocationID to InventoryLocationID

Representative performance for a sampling of web services is illustrated in Table 2. Overall, the system erred on the side of making false positives and was able to maintain a matching accuracy in the range of 75-85%.

4. CONCLUSIONS

In this paper, we have presented an approach to semantically match two API schemas to enable the chaining of their associated services. Building automation into this task enables scalable deployable solutions in the world of internet where the web services are being added at a rapid pace.

4. REFERENCES

[1] T. Berners-Lee et al. The semantic web. Scientific American, 2001.

[2] A. Goldberg and Kennedy. An efficient cost-scaling algorithm for the assignment problem. SIAM Journal on Discrete Mathematics, 6(3):443{459, 1993.

[3] J. Blythe et al. The role of planning in grid computing. In Proc. ICAPS, 2003.

[4] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In The VLDB Journal, pages 49-58, 2001.

[5] D. McDermott. Estimated regression planning for interactions with web services. In Proc. AIPS, 2002.

[6] S. Melnik et al. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In Proc. ICDE, 2002

[7] G.A. Miller. Wordnet: A lexical database for English. Communications of the ACM, 38(11):39{41, 1995.

[8] Evren Sirin, James Hendler, and Bijan Parsia. Semi automatic composition of web services using semantic descriptions.

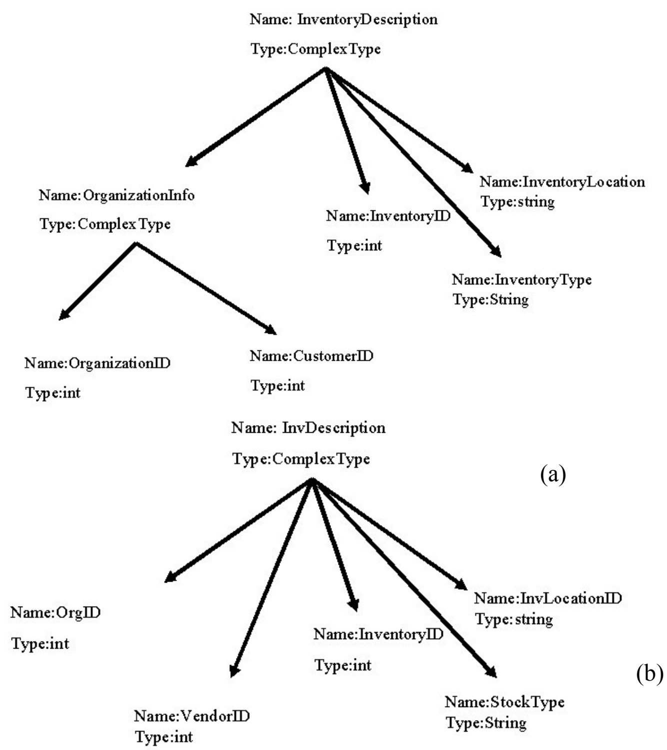

Figure 2: Illustration of semantic matching between APIs for web services exchanging business objects.

S. No.	Source attribute	Matching destination attribute	Matching Score	Contributions in order
1.	OrganizationID	OrgID	2.5	0.67,1.0,1.0, 0.50
2.	InventoryLocation	InvLocationID	3.0	0.74,0.67,1.0,1.0
3.	InventoryID	InventoryID	4.0	1.0,1.0,1.0, 1.0
4.	InventoryType	StockType	3.0	0.56,1.0,1.0, 1.0

Table 1: Matches produced by semantic match for the pair of services ADTs shown in Figure 1.

S.No	Source ttributes	Destination attributes	Correctly matched	Missed matches	Spurious matches	Actual matces	% Accuracy
1.	10	15	8	1	2	9	81%
2.	23	34	28	3	7	31	81.57%
3.	67	73	29	5	9	34	79%
4.	84	56	10	3	4	13	76.4%

Table 2: Illustration of performance of semantic matching during chaining of services derived from business objects.