It seems to me the natural formulation of Intent solving is in the form of Constraint Satisfaction Problems, since we're doing counterparty discovery for the purpose of optimizing matching in respect to what any entity wants to give and get, i.e. finding good matches under constraints.

Constraint Satisfaction Problems (CSP)

One way to formulate an Intent

is as a [Constraint Satisfaction Problem].

It is a very natural formulation, since we can encode Predicates

over Resources

as Constraints

over Variables

. This approach enables us to use results from a rich research history and it has interesting special cases, e.g. Linear Programming.

CSP Definition

A CSP consists of the following sets:

- $X = \{X_1, \ldots, X_n\}$

its Variables

- $D = \{D_1, \ldots, D_n\}$

the Domains of Values for its Variables

- $C = \{C_1, \ldots, C_m\}$

its Constraints

Any variable X_i

can take values from its corresponding domain D_i

.

Constraints are of the form $C_j = \langle t_j, R_j \rangle$

, with t_j

being a subset

of X

of size k

and R_j

a k

-ary relation

over their corresponding domains.

An evaluation

of variables is a mapping from a subset of X_i

to values from their D_i

. A constraint C_j

is satisfied

by an evaluation if the values of the variables t_j

satisfy the relation R_j

.

An evaluation is consistent

if it satisfies all constraints and complete

if t_j = X

. An evaluation that is consistent and complete is called a solution

which solves the CSP.

Correspondence of Terminology

Turning a Transaction

containing a set of Partial Transactions

, which in turn contain input and output Resources

into a CSP gives us the following correspondences of Terms:

- Variable: A Position

for a Resource

in a Predicate

.

- Domain: Restrictions

on which Resource

s can fill a Position

. Restrictions

are: Resource Type

, Input or

Output Resource

in a PTX

, ephemeral or

non-ephemeral Resource

.

- Constraint: Each Resource

containts a Resource Predicate

which defines the required relations between Resources

inhabiting Positions

.

- Evaluation: Once Positions

are inhabited with Resources

which fulfil the Restrictions

, evaluate all their Predicates

.

- Solution: All Predicates

from all the above Resources

evaluate to True

.

Example: The three coloring Problem

In a fully connected graph G

with vertices $v \in V$

, edges $e \in E$

each vertex is supposed to be painted in a color $c \in {Blue, Red, Green}$

with no v_1, v_2

sharing an edge having the same color.

Formulated as a CSP we get:

- $X = {v_1, \ldots, v_n}$

as Variables

- $D = {Blue, Red, Green}$

as Domains for each Variable

- $C = {C_i = \langle {v_i, v_j}, v_i \neq v_j \rangle }$

, a set of Constraints, requiring pairwise inequality for neighboring vertices

Enter Intents

To close the bridge to Intents

, lets assume:

- $A = {A_1, \ldots, A_n }$

is a set of Agents who want to collaboratively color a graph.

- Resources exist that encode specific Vertices via the Resource Type

and specific colors via the Resource Value

.

- All Resources

are non-ephemeral

- The required Vertices are Input Resources
- Output Resources

are the same vertices with signatures proving they were part of a valid three coloring.

- Every Agent owns some of these Resources
- Every Agent wants to spend exactly one Resource

. This is an additional Constraint, to be encoded in the Resource Predicate

along with the inequality.

Vertex ownership example:

- A_1

owns v_1^{Blue}, v_2^{Red}

- A_2

owns v_1^{Red}, v_1^{Green}, v_1^{Blue}

- A_3

owns v_3^{Blue}

The agents would submit this to a solver, which would find the solution of (A_1, v_2^{Red}), (A_2, v_1^{Green}), (A_3, v_3^{Blue})

by performing search over possible evaluations.