

Web Services for Reflective Learning¹

Simon Grant, Adam Marshall, Janet Strivens and Roger Clark

Version of 2004-11-1

Keywords: *Personal Development Planning (PDP), Personal information management, Personal Learning Environments (PLE), Reflective learning, web services, JISC ELF Common services layer, e-learning interoperability, IMS Learner Information Package, IMS Learning Design.*

¹ This document constitutes Deliverables d1.1 – d1.8 of the JISC-funded WS4RL project, <http://www.jisc.ac.uk/index.cfm?name=ws4rl> and http://cetis.ac.uk:8080/frameworks/learning_domain_services/personal_development_management/ws4rl



WEB SERVICES FOR REFLECTIVE LEARNING.....	1
OVERVIEW	3
1 A MODEL OF PERSONAL THEORY BUILDING AS PART OF PERSONAL DEVELOPMENT. 4	4
1.1 A BASIS FOR UNDERSTANDING SOME ASPECTS OF PDP	4
1.1.1 <i>Personal Theory-Building</i>	4
1.1.2 <i>Personal theory-building and formal education</i>	5
1.1.3 <i>Personal theory-building and the self</i>	6
1.1.4 <i>Personal theory-building and Personal Development Planning (PDP)</i>	7
1.1.5 <i>Background survey of PDP practice</i>	7
1.2 A MODEL OF THE PROCESS	8
1.2.1 <i>Purpose and nature of the model</i>	8
1.2.2 <i>Noticing</i>	8
1.2.3 <i>Documenting</i>	9
1.2.4 <i>Recollecting</i>	10
1.2.5 <i>Theorising</i>	12
1.2.6 <i>Goal setting</i>	12
1.2.7 <i>Action planning</i>	13
1.2.8 <i>Acting</i>	14
1.3 A SCENARIO FOR PERSONAL THEORY BUILDING	14
1.3.1 <i>The scenario</i>	15
1.3.2 <i>Reflections on the scenario</i>	15
1.3.3 <i>A simple version</i>	16
1.3.4 <i>A richer version</i>	16
2 WEB SERVICES FOR REFLECTIVE LEARNING: FUTURE POSSIBILITIES.....	17
2.1 THE SYSTEMS INVOLVED IN THIS DISCUSSION	17
2.1.1 <i>Personal Learning Environment (PLE)</i>	18
2.1.2 <i>Personal Information Aggregation & Distribution Service (PIADS)</i>	19
2.1.3 <i>PDP web service (PDPWS)</i>	22
2.1.4 <i>Repository of learning materials and designs</i>	23
2.1.5 <i>The databases</i>	23
2.1.6 <i>A note about security, authentication and authorisation</i>	23
2.1.7 <i>The relationship between PIADS and a Shibboleth Attribute Authority</i>	26
2.2 FIRST USE CASE: A SMALL PDP ACTIVITY EMBEDDED IN A SHORT COURSE	28
2.2.1 <i>Web services required for the first use case</i>	29
2.3 SECOND USE CASE: LEARNER-DIRECTED PDP	30
2.4 WEB SERVICES CALLS AND RETURNS	32
2.4.1 <i>PIADS calls</i>	32
2.4.2 <i>PDPWS calls</i>	36
2.4.3 <i>Generic process for learner-directed use of a PDPWS</i>	44
2.5 A FEW POSSIBLE NODES IN A PDPWS MAP	48
2.5.1 <i>Learning log or journal</i>	48
2.5.2 <i>Review of level of confidence in a skill area</i>	48
2.5.3 <i>Setting goals</i>	48
2.5.4 <i>Discussing an action plan with a learning advisor</i>	49
2.5.5 <i>Planning an essay</i>	49
3 ACKNOWLEDGEMENTS.....	49
4 REFERENCES AND BIBLIOGRAPHY FOR SECTION 1	49
APPENDIX A: ACTIVITY DIAGRAM FOR PERSONAL THEORY BUILDING	51
APPENDIX B: A MODEL OF PROVIDING AN ESSAY PLAN BY WEB SERVICES	54
APPENDIX C: GENERIC TYPES	55

Generic activity types 55
Generic output part types..... 56

Overview

This document firstly presents some philosophical ideas on reflective learning as part of personal development planning (PDP), and secondly proposes some related web services (WS) that might serve in the cause of PDP.

We are well aware that PDP cannot be easily specified or modelled normatively (any more than can education or learning in general). Rather than try to offer an overall model of PDP, we set out a model of one possible process which we call “personal theory building”. This process at least conforms to (i.e. fits within) the definition of the nature of PDP arrived at by the EPPI group which undertook a systematic review of research relating to the effectiveness of PDP (Gough *et al* 2003). (This group agreed that PDP must include reflection together with at least one other process out of recording, planning and action.)

We are then able to detail and formalise the process of personal theory building, partly so that it can in principle be represented using interoperability specifications such as Learning Design (LD), and also in order to show how this can be related to previous categorisation of PDP activities done in the survey mentioned in section 1.1.5. Sample scenarios are presented and explained in UML.

The main purpose of formalising such a process is to prepare the ground for it to be offered as a web service. The second half of this document takes this forward, by setting out a future architecture for web services in PDP, detailing some of the most important web service calls, and discussing and illustrating how web services would be used to implement the two main scenarios of use of PDP: firstly where PDP is embedded in other learning, and secondly where it is free-standing. Some examples from the first half are brought in, to illustrate how this could be done for typical PDP processes.

PDP often involves reviewing previously stored records and creating new ones – including information about educational and other activities, achievements or qualifications, skills or competencies, goals or aspirations, interests, and the products of and evidence for many of those things. These same records also relate to e-portfolio usage, and selections could be presented to other people for a variety of purposes. Recognising this, central to the presented architecture is the factoring out of records-related functionality from PDP process-related functionality. We identify a major architectural component as the Personal Information Aggregation and Distribution Service (PIADS), which is used by the PDP web service, but may also be used in other contexts.

The purpose of the PIADS is to give learners, as owners of information related to themselves, the facilities to manage the storage of and access to that e-portfolio-related information. Learners will be able to choose from possibilities of where particular information is to be stored, and who is allowed to access it. As PDP is closely tied to much of this information, it makes sense for any PDP web service to use through the same PIADS service.



Again, because of the same close relationship of PDP and personal information, one cannot specify in detail what PDP services can be offered independently of the information about the learner. Even in an e-learning context, the PDP interaction has to be a dialogue, with the learner asking to do some PDP, and the PDP services asking more about the learner to determine what is appropriate.

1 A model of personal theory building as part of personal development

Increasingly, education in particular and learning in general is recognised as a process which can (perhaps should) be complemented by more general personal development, aimed at helping learners to be more autonomous and self-directed, and more able to choose, direct, manage and evaluate their learning throughout life. The role of what has come to be known in UK higher education, and some other circles, as “personal development planning” (PDP) is thrown into sharper relief by the currently evolving context of learning. Pressure on resources seems to be leading to a reduction in the support that traditionally came from an educational institution. The development of e-learning, which is one possible response to this pressure, also comes along with the idea that learning can take place, more easily than before, outside the context of an institution or other body. How are the vital PDP processes going to proceed in these new situations?

In response to this, firstly electronic tools and now web-based tools have been emerging from several sources which are intended to help support PDP processes. “Virtual Learning Environments” (VLEs) such as Bodington² now hold learning materials electronically, together with support for some of the processes which use or complement the materials, and these are intended to lead to more effective learning than plain self-study. The question then arises, can web-based PDP tools (such as the soon-to-be open sourced LUSID³ system,) be integrated with e-learning, including developments from Virtual Learning Environments (VLEs) and Personal Learning Environments (PLEs)?

Just as effective development of e-learning tools is helped by a clearer understanding of the learning process, so PDP tools need a clearer understanding of PDP processes. Section 1 of this paper approaches the desired clarification in two main sections: Section 1.1 explains a basis for understanding one of the core aspects of PDP; Section 1.2 proposes a model of this core. As it is essential for this model to be grounded in recognisable practice, a modified UML activity diagram is given in Appendix A and related to a scenario which is given in Section 1.3.

1.1 A basis for understanding some aspects of PDP

1.1.1 Personal Theory-Building

Our basic model owes much to constructivist psychology, with perhaps the key influences being the work of George Kelly (and Lev Vygotsky). Kelly (1955) used the metaphor of the

² <http://bodington.org/> and <http://sourceforge.net/projects/bodington/>

³ <http://lusid.liv.ac.uk/>

scientist to emphasise how all human beings attempt to make sense of and explain to themselves the external world. This is well-expressed in this quotation from a relevant web site (<http://mind-brain.com/personality/kelly.php>).

“They have constructions of their reality, like scientists have theories. They have anticipations or expectations, like scientists have hypotheses. They engage in behaviours that test those expectations, like scientists do experiments. They improve their understandings of reality on the bases of their experiences, like scientists adjust their theories to fit the facts.”

Kelly stressed that this is not purely, or even primarily, a cognitive process: it includes perceptions, emotions and behaviour.

Here we call this process “personal theory-building”. It is personal because each individual’s theories will differ according to the experiences they are built on, and the uniqueness of each individual’s process of sense-making. It is theoretical because it seeks to generalise and explain, in order to help us to predict future events. In order to generalise, we need to employ the ability to analyse, and abstract certain aspects of the whole experience: this allows us to perceive (or create) patterns and connections between experiences.

Much of this pattern-noticing goes on below the level of consciousness. We think of this as peculiarly true of infants and young people, but a human being at any age can extract the patterns from experience which will direct future behaviour without being aware of this process. We are more likely to become aware of the process when our predictions fail – we meet new experiences which do not make sense in terms of our existing theories. This may be a positive or a negative experience – we may become anxious or aggressive, or feel excited by the possibility of new insights. Either way, we will probably **notice** the experience/event insofar as we could recount it to another and consciously recall it at a later time. It has some significance for us.

1.1.2 Personal theory-building and formal education

Formal education could be characterised as the deliberate stimulation of personal theory-building. Vygotsky talks much of teacher’s role in the “scaffolding” of experience: this encompasses both the selection of experiences for the learner to try to make sense of, thus stimulating the further elaboration of the theories s/he has built so far; and also the bringing into consciousness of this process, by making sure the experience is noticed and explicitly encouraging the learner to theorise about it. Because there is an outside influence – the teacher – guiding the process, there may not need to be the same strength of motivation in terms of affect associated with the experience in order to stimulate the theorising. Typically the learner is told how to analyse the experience, what aspects to abstract, what connections to make and what explanations to accept. This works for most people quite a lot of the time, and certainly makes the process of education more efficient than a regime based totally around self-discovery. However, it has its drawbacks, in that the learner may not feel personally motivated to test the theories against new experience. Inconsistencies may not be noticed, and indeed, what teachers would call “misconceptions” may be held.

Certainly by the time the learner reaches tertiary education s/he should be developing less dependence on the “scaffolding”. In higher education, ‘deep learning’ has been a persuasive concept guiding pedagogic innovation (see the body of work by Entwistle *et al.*). Deep learning assumes an active role for the learner in seeking out connections and testing out ideas. Abstract thinking is valued and the process of theory-building is made explicit. Although the pace of this varies somewhat across disciplines, the learner is expected to be able to articulate personal theories and subject them to careful examination in a public way, comparing and contrasting them with the personal theories of others. (Some educational environments have a “hothouse” effect, cosseting learners into performance without understanding, producing people who can do things without being aware that they can do them or knowing what it is that they can do. Some assessment regimes reinforce this effect, producing what has been called ‘surface learning’, which is seen as ultimately deleterious to lifelong learning.)

Academic tutors remain responsible for the selection and presentation of experiences in traditional academic courses, but increasingly students in higher education have the opportunity to learn from the workplace, on clinical placements, work-based learning modules, sandwich years and so on. Here the learning experiences by definition are not carefully selected and paced: on the contrary, the learning environment is rich but chaotic. It depends on the learner noticing things, recognising them as significant in some way, analysing them and comparing them with previously-formulated constructs. It is assumed to be helpful if this process is made explicit, partly so that the tutor can check on the learning, but also because being explicit helps the theorising process for the learner. Hence the ubiquity of reflective journals/learning logs in relation to such experiences.

1.1.3 Personal theory-building and the self

So far, we have been discussing the role of personal theory-building in relation to the acquisition of discipline-based knowledge and understanding. However, the debt to Kelly should remind us that we are ourselves a key object of our theory-building. Individuals may vary in the amount of navel-gazing they do, but we all form a view of ourselves, our likes and dislikes, our skills and personal qualities, from the ways we are aware of ourselves reacting to life’s experiences. It does seem that there is more scope for this part of the process to ‘go wrong’ in the sense that we can form conceptions of ourselves and how we will act/react to new situations which are wildly out of kilter with the theories of those around us who also observe our actions. This is predictable if we consider that our learning and theorising about ourselves is rarely scaffolded for us by a sensitive teacher, still less made explicit and compared with others’ theories about us in a systematic way. Personal theory-building about oneself almost inevitably involves affect, as we are bound up in the experiences about which we are theorising.

What would a pedagogic support to this process look like? Many of the possible techniques are familiar from other fields. There are some popular tools, in the form of standardised tests which put together responses to questions about past or hypothetical experiences to give a predictive profile. The concept of a mentor or ‘critical friend’ has wide currency, referring to someone who will help one ‘debrief’ an experience, in terms of standards of performance, actions taken and their consequences and, crucially, the emotions experienced. A ‘life coach’ helps an individual think through personal goals and motivations to plan future actions. Self-

report and self-assessment is balanced by multiple perspectives on an experience from other participants.

1.1.4 Personal theory-building and Personal Development Planning (PDP)

All of the above techniques and more are incorporated within an approach to supporting personal theory-building about the self, called Personal Development Planning (PDP). PDP is defined as “a structured and supported process undertaken by an individual to reflect upon their own learning, performance and / or achievement and to plan for their personal, educational and career development.” (QAA Guidance document on Progress Files: <http://www.qaa.ac.uk/crntwork/progfileHE/contents.htm>) From the academic session 2005-6, all Higher Education Institutions (HEIs) in the UK will be expected to provide opportunities for this process with all students at all levels of their programmes.

As suggested above, most personal theory-building in academic institutions is about a subject discipline. In PDP the subject and object of personal theory-building is oneself, the learner. Broadly, PDP activities can be categorised into those which look back on and analyse past experiences, and those which look forward, planning the next step or the ultimate goal. They are rather different sorts of activities and a learner may need to be supported to do each separately. They could be likened to the difference between hypothesis-building and hypothesis-testing. Some individuals find one easier and some the other, but both need to be undertaken if knowledge is to advance – in this case, the individual’s self-knowledge.

At the centre of these processes – between backward and forward looking, or between reflection and planning – stand learners’ theories of themselves. The aspect of PDP on which we are focusing relates closely to these central theories. While effective learners may carry out the processes intuitively, and build theories unconsciously, it often falls to the educator or PDP practitioner (or mentor, supervisor, life coach, friend, therapist) to make the processes explicit; or where they remain implicit, to offer appropriate guidance, direction and information so that the learner is more easily able to arrive at a theory which will be truly useful.

1.1.5 Background survey of PDP practice

Our understanding of the nature of PDP is informed by direct experience of current practice; but also by a survey which started in 2003 as part of the work of the Centre for Recording Achievement (2004). The survey covered a range of PDP-related practice in educational institutions, including some schools as well as the core of further and higher education in the UK. The survey is situated on a web site which is still being added to. The analysis on the site starts with a top-level description of the PDP programme surveyed, and extends right down to the point at which each element of a PDP process is characterised in terms of a set of generic types of activity; while the outputs from those processes (generally written records, whether kept on paper or electronically) are characterised in terms of a set of generic output parts (see Appendix C).

These generic activities and parts were developed over the course of the initial survey, ensuring that the categories made sense to the PDP practitioners. Where any practice was found which was not adequately represented by the activities and outputs which had been

present up to that time, one or more new categories were devised to represent that practice. In the end, the range of practice surveyed, together with the fact that new surveys were not turning up new categories, suggested that we had captured a reasonably full set of activities and outputs characterising PDP process as a whole.

As these categories are taken from real practice, they are not geared to fit in with any theoretical approach to PDP. So we decided to relate them to the model which is now to be laid out, as a kind of “reality check”: we wanted to be sure that we talk about processes related to PDP as it is currently practised, not an imaginary process divorced from real life.

1.2 A model of the process

1.2.1 Purpose and nature of the model

A major stimulus for creating a model covering reflective practice in PDP is to test out the idea that such processes can be represented formally. A formal representation is needed to work towards the offering of PDP as a web service, by any educational institution, employer or other body wishing to support PDP among people that are affiliated to it.

For this purpose, it is not necessary to try to detail all the practices and coverage of the whole area of PDP. In any case, there is likely always to be as much scope for diversity and innovation in PDP as in any other educational process. What is required is to illustrate a reflective process in sufficient detail both to enable readers to get a good sense that this is indeed the kind of process which is undertaken in PDP, and to test out in practice the a formalism to use for the representation of the process.

Because of the needs of technology projects we are involved with, our particular interest involves representing PDP processes in terms of a suitable technical specification, and representing the learner information which is fed into a PDP process, and also that which results from the PDP process, using an information model similar to the IMS Learner Information Package (<http://www.imsglobal.org/profiles/>). The concept is that where the learner stands, from the point of view of PDP, is effectively what is represented in their personal learner information or learner profile, and that any effective PDP process should modify that profile in some way. The contents of the learner profile can potentially be used to determine any choices in the PDP process which is undertaken.

This PDP process model is then to serve as an exemplar and prototype for other PDP processes to be represented similarly, while not being in any way normative with respect to the contents. The level of detail of the model needs to be suitable for representation as a UML activity diagram. As the model needs to relate closely to PDP, and to be recognisable to PDP practitioners, so we have listed, with each major step of the model, the related generic PDP processes and outputs taken from the above-mentioned survey.

1.2.2 Noticing

There are many reasons why a particular experience may come to the attention of a learner, and these could be roughly categorised as affective, guided or serendipitous.

Firstly, an experience may have a large affective component, perhaps delightful or traumatic. Experiences of this kind are sometimes the subject matter of dreams as well as discussion and conscious reflection. There is a clear potential utility for reflection on this kind of experience – most learners have an interest in maximising pleasure and minimising pain.

Secondly, attention may be drawn to an experience through a process of guidance or support. In the PDP context, the learner may be expressly asked to record experiences in a learning log or journal, and the guidance for recording these experiences may include indications about what aspects of the experiences are to be thought of as significant. The guidance may be communicated through rubrics on forms, on paper or electronic, or it may be communicated in person, either by way of formal instruction, or through less formal conversation and discussion.

Thirdly, a learner who is prepared to look out for significant things may just notice something of interest, even where there is no guidance given to notice it, and where it has no direct affective connection. This might be the easily-recognisable case of noticing something that is odd, strange, out of place – perhaps motivated by curiosity, or an eye for order and pattern. We could call this kind of noticing “serendipitous”, as it could be seen as potentially useful accidental discovery.

As an aside, the question may be raised about what we mean when we refer to experiences that are noticed. What is it, exactly, that is noticed? We assume that there is no single definition of a whole experience, but rather that particular aspects of the experience are picked out as significant from among the things which are perceived and remembered, whether or not perceived and remembered consciously or correctly. It is this personal representation of an experience that is referred to whenever the word “experience” is used here.

The learner is obviously the key actor in this activity. Another person in the role of helper, tutor, mentor, etc. can also take part, by drawing the learner’s attention to particular experiences or aspects of experiences. In the survey, there is one generic PDP process which is directly related to this step.

- Discussing learner's personal situation / experiences

Presumably, the reason why not many generic PDP processes appear under this heading is that a supported process is more possible when there is documentation, rather than simply undocumented noticing. In respect of the information involved, noticing precedes the stage at which there is tangible output. The immediate output of noticing is just some kind of memory trace. The act of recording what is noticed brings us on to the next step, documenting.

1.2.3 Documenting

The content of the experience needs to be captured in some way so that it will be able to be made available to the reflective processes of the learner, as well as to any ICT tools used in conjunction with PDP. This documentation may be done spontaneously, as with a keen diarist, or it could be stimulated or prompted in a number of ways. One way is through the ICT: a system may automatically prompt the learner to record something. Another way is through discussion with another person, maybe one in the helper role.

Not only does recording an experience greatly change what can be done with ICT tools, but the very fact of committing something to paper-based or electronic storage may influence what is remembered, and even how the activity is undertaken in the first place.

For our purposes, we group together the documenting step along with the guidance on what to notice, and any prompts or rubrics for documentation. Our model of documenting will therefore represent such guidance or prompts.

The earlier survey came up with these generic PDP processes, which relate to this step.

- Compiling list of experiences or past activities, including employment
- Listing achievements / qualifications (with documentation if available)
- Writing log (for learning or reflection)

At this stage, we imagine the process of documentation by itself to be as free of theory as possible, so that it can be open to influence from the learner and his or her individual situation, and so that the output can reflect, more or less closely, what is in the learner's mind.

Possible generic outputs that may be associated with the step of documenting include the following, taken from the same survey. These do appear to be relatively less dependent on the theory used.

- Statement of learner's personal situation
- Statement / list of past activities / experiences / employment
- Reflective writing, referring either to experiences or to journals / logs
- Statement / list of other achievements
- Statement / list of formal qualifications
- Learning log or journal intended for later reflection

Our model of documentation is thus the committing of experience to a form that can potentially be accessed and retrieved independently of the learner's own memory. There will typically still remain much extra detail in memory that was not committed to record, but one purpose of such records is to avoid the gradual internal reworking and changing of memories (Bartlett, 1932): what has been recorded should be sufficient to prompt a realistic recollection of the experience at the time.

In practice, the very fact of documenting experience could well provoke further recollection and theorising, but for the purposes of our model, we will regard the documenting step as separate from those further ones.

1.2.4 Recollecting

We have characterised the stage of documenting an experience as being relatively free of theory; but when it comes to recollecting, theory is much more likely to be playing a part. A natural and spontaneous recollection might occur when a person might think, perhaps "ah now, let me see, that reminds me of something that happened last month...". Or perhaps the person recollecting might be thinking of a series of past experiences. Whichever way it is, some significant association is going on, some connection is being made, and we can imagine the natural theory-building processes as in train.

In contrast, and more relevantly for the pedagogy of personal development (or psychotherapy or life coaching), a mentor, or a mentoring process, might prompt such recollection. “How were you feeling when that happened?” “What happened last time you tried that?” “Were you feeling the same when this happened before?” “Think about all the times you can recall when you have been presenting to an audience.” All these, and similar questions, are likely to be ways in which another person tries to bring to learners’ attention sets of evidence which it is believed might be effective in promoting learning about themselves – helping their personal theory-building.

These processes are perfectly possible with a skilled helper who knows about the learner, without referring to records. But the presence of documentation can both help helpers know what to ask, or help learners to put together significant collections of experience without the explicit help of another person. So what is likely to happen with documentation? We can start from related generic activities from the survey, which are as follows.

- Reviewing and reflecting on logs
- Reviewing experience in response to guidance
- Reviewing coursework performance and course experience
- Reviewing critical incidents
- Assembling evidence for skills
- Originating CV / personal statement / other compilation
- Revising CV / personal statement / other compilation

What we have here are descriptions of processes which may run through into theorising, but the theorisation is not yet explicit in the process. The influence of some kind of theoretically-related outcome is apparent, though. Leaving aside the spontaneous process, which is not yet well-understood, either the helper (mentor, tutor, etc.) has an implicit or explicit idea about the kinds of evidence which will influence the learner towards a desired outcome, or the system (paper or interactive) is designed with this kind of outcome in mind – the theory being in the minds of those who designed the system. Each case illustrated by the above processes can be related to a developing internal theory of the learner him or herself. What can he do? How can she conceive of her skills? What next steps are appropriate for further development?

Here are the generic outputs from the survey that can be related.

- Statement of performance on task or goal
- Evidence for skills
- Statement / list of personal interests
- Statement of learner support provided (past) or required (future)
- CV / personal statement / other compilation of above items

Again, these outputs might include some personal theory, and it is difficult to draw a line between the kinds of output which does include such theory and that which does not. The case of CVs perhaps merits special attention. Though good practice would suggest that a CV should embody explicit personal theory, the formulaic nature of much CV writing is more in keeping with merely putting information together.

For our modelling purposes, it is convenient to make the distinction at that point where theory gets involved. Recollection is the marshalling of evidence that is likely to be relevant to

personal theory-building, either spontaneously or following the guidance of those who have an idea of what evidence might be relevant.

1.2.5 Theorising

Once the captured experiences, assumed to be relevant, have been brought back to mind, the stage is set for the learner to form or modify theories which help to explain the significant features of the experiences. From our point of view of personal development, we are most interested in the learner's theories about aspects of the learner him or herself, which may have affected the experience.

In the generic activities drawn from the survey, there is sometimes only a shading between these activities and the previous ones which formed part of recollecting. Most of these activities could be understood to include the step of recollecting as well as that of theorizing.

- Reviewing past written goals and action plans against more recent past experience
- Relating experiences to skills (or vice versa)
- Reviewing / profiling / auditing skills
- Reviewing progress in / development of skills
- Reviewing personal interests
- Reviewing / reflecting on personal attitudes / values
- Assessing own learning style
- Relating goals to motivations and reasons

What makes the difference in our view is that these activities do suggest the likely presence of theorizing. Having a conception of one's own skill in a particular area is one form of holding a personal theory. Knowing how effective one has been in achieving one's goals helps build a theory of personal effectiveness – and might lead to a revision of goals and plans; and so on. The following outputs from the survey also suggest the presence of personal theory, and they might be expected to result from these activities.

- Statement of learning from experience
- Statement / list of skills
- Evaluation of skills on a scale
- Identified strengths and weaknesses in skills
- Statement of learner's attitudes / values
- Statement of learning preferences
- Statement of reasons for present module / programme / position
- Statement of motivation / aspirations for future position / employment / course

Our model of the theorizing step thus starts from a set of recollections present to mind, or to hand, and finishes with a personal theory, or a newly-enhanced personal theory.

1.2.6 Goal setting

The next step in the model process needs to involve the use of the newly-enhanced personal theory. In general PDP practice, this next step is often setting goals. There are at least two ways in which setting goals is dependent on personal theory. Firstly the goals need to be

based on persistent personal theory for them to be acted on consistently. If, instead, goals were based on the feeling of the moment (however induced) they could well never be followed through, as a later checking of feeling against the goal might not fit. Secondly, the personal theories on which the goals are based need to be realistic. If they aren't, the learner runs the risk either of underachieving, or of overreaching, with consequent failure – which, however, could be used in the next round of personal theory enhancement.

There is a clearly defined set of generic PDP activities which correlate with this step.

- Setting goals for skills development
- Setting goals related to subject development
- Setting more general personal / social goals

The outputs are closely related.

- Statement of current goals without stated motivation
- Statement of current goals with stated motivation

The latter output could be seen as involving aspects of the personal theory along with the goals. The goals can involve theory either because the plausibility of the goals depends on the truth of the theory, or because the goals are to remedy undesirable states that have been revealed by the theory-building.

Goals can be set spontaneously by learners, or alternatively the person in the helper role can help by discussing possible goals, and the suitability of goals. This may involve helping the learner to take account of the personal theory fully in the course of goal setting. In general, ICT systems have a limited role in the choice of high-level goals. One example is in selecting career areas (Graduate Prospects Ltd., 2004). A clearer role for ICT systems is in supporting action planning.

1.2.7 Action planning

In the survey of PDP practice, there were several other generic activities noted that seem to follow on from the steps of theorizing and goal setting. The common follow-up to goal setting in the PDP context is action planning, which is the detailing of sub-goals and planned activities which are designed to lead to the achievement of the set goal. As with goal setting, action planning uses the enhanced personal theory, this time not just to set appropriate goals, but to select means of achieving those goals that are plausible given that personal theory.

- Originating action plan for the achievement of academic goals
- Revising action plan for academic goals in the context of feedback / discussion
- Originating action plan for personal / skills development / goals
- Revising action plan for personal goals in the context of feedback / discussion
- Choosing / evaluating suitability of course / module / employment / position
- Writing application for position / employment / course / programme
- Writing individual learning plan
- Negotiating learning / employment contract

Action planning is typically not something that is spontaneously made explicit or documented. Discussion, or sharing the action planning process, with someone experienced in

action planning can help to stimulate the making of explicit plans. And there is also a great potential in ICT tools to help with this step. Low-level tools include personal organisers and diary or calendar systems. Then there are various project planning tools already available. The real potential for action planning tools is, however, not just when they facilitate action planning as such, but when the actions planned are then made available for the next round of reflective processes, where the planned actions can act as a prompt for documenting the salient things that are noticed about the actual actions that took place, as well as the actions as planned being suitable objects for recollection, and comparison with what happened in practice.

The action planning steps above may typically result in generic outputs noted in the survey.

- Action plan where learner not directed to particular type of goal
- Action plan towards next life stage (position, employment or study)
- Action plan for skills development
- Action plan for academic /educational / work-related goal(s)
- Action plan for personal / social goal(s)
- Learning / employment contract (include mention of skills)
- Individual learning plan

These are still action plans, at the level of intentions – the actual achievement of the goals that have been set the is not a part of theory-building, but rather the working out in practice of the action plans built on the theory. Thus, the final output of our model of personal theory-building can be seen as the theories in the mind of the learner, together with relevant goals and planned actions based on that theory, as appropriate for the learner.

1.2.8 Acting

This step belongs more to a model of an experiential learning cycle than to the model of personal theory building itself. We see actions both as the end result of putting personal theory into practice, and as the raw material from which observations can be made, and things noticed. And acting is an important consideration in PDP as well, where it is sometimes explicitly written in to the programme in terms of developing skills.

- Doing exercises alone for skill development
- Participating in workshops / classes / sessions for skill development

Of course, such exercises are themselves proxies for real world experience, designed and guided by the teacher, who will also typically guide the analysis of the experience and highlight aspects for the learner to reflect on. Real world actions will be very varied, and in any case not properly part of PDP but part of the life that is helped through PDP. Thus, there are no generic outputs at this stage, as any output would depend entirely on what the activity is.

1.3 A scenario for personal theory building

The model which above has been outlined in words needs to be formalised sufficiently to represent the structure of a supported PDP process. We present here a UML activity diagram which relates to the model as a whole (in Appendix A), and a scenario in three different

versions, which can be seen to relate directly to the diagram. Capital letter references in the text refer to the same letters in the diagram.

1.3.1 The scenario

The learner is in a work situation, where two colleagues give presentations of their current work to the team. The learner enjoys [A] one of the presentations much more than the other one [D], and asks for copies of the handouts to help later recollection for personal study. That evening, she makes a note in her learning log [H], and annotates this note as relevant to communication and presentation skills.

The following week, knowing that she is going to have to make a presentation in her workplace, she compares her recollections of the two presentations with the last two she gave [L], and looks up the guidelines which are available. She re-assesses the level of her own presentation skills against the guidelines, and makes a judgement of where she is in terms of presentation skill [N].

The learner consults her mentor, who agrees that this is an appropriate issue to focus on at this stage [P], given her imminent presentation task, and a goal is set to improve the presentation style [Q]. She plans appropriate actions to achieve this. The guidelines help her to focus on different aspects of her presentation. Analysing the presentations she has watched and made, she decides to focus on better quality and use of audio-visual aids. She plans her forthcoming presentation [S] referring back to the two presentations she attended recently, analysing them specifically in terms of the quality and use made of audio-visual aids.

The presentation goes well [Z], she appreciates the positive reactions from audience and mentor [A, D] and records this in her log [H]. Because the individual action was anticipated, the recollection that goes on as an integral part of recording [L] is sufficient, without needing any further recollection, for her to know that she can now use audio-visual aids better in presentations [N]. Privately, she knows that whenever another presentation comes up, she will be able to go through the same process with more confidence.

1.3.2 Reflections on the scenario

This scenario given above is intended to be naturalistic, and in the form above does not explicitly involve any ICT systems in its operation. It is a plausible, if idealised and stylised, example of a process which should be recognisable by most people involved in helping learners with this kind of skill development, whether in the context of formal education, training or work.

In order to make it directly relevant to the kind of PDP systems that use ICT, we will diverge in two directions: a simple version in which the possible role of an existing ICT system is evident; and a richer version in which more advanced systems are imagined. The base scenario above gives context for the understanding of the simple and richer versions, which are based on the same situation.

1.3.3 A simple version

The learner is in a work situation, where colleagues give presentations of their current work to the team. The learner enjoys [A] one of the presentations much more than the other one [D]. Towards the end of the day, completing her CPD records, the system she is using guides her [G] for an entry in her learning log [H], and this is enough for her to recollect previous experiences and assess her current level of competence in presentation [N]. In the CPD system, the learner sets a goal to improve her presentation style [Q]. She uses guidelines from the system [R] to plan her forthcoming presentation [S]. (*The execution of the plan is omitted, as action is not part of the main model.*)

1.3.4 A richer version

The learner is a student of marketing on a work placement, and two work colleagues are to give presentations of their current work to the team. As the team meeting has been put in the student's appointments system, as this information is available to her PDP system, and as presentation skills feature in the subject benchmarks for her course, the previous evening the learner is reminded about the presentations, and it is suggested that she attends to various details [C]. The learner recognises one of the presentations as more effective than the other one [D], and asks for copies of the handouts to help later recollection for personal study. That evening, her PDP system reminds the learner [G] to enter reflective notes of the experience in her learning log [H], and to analyse this in terms of its relevance to communication and presentation skills.

The following week, her PDP system has noted the task of making a presentation in her workplace, and reminds the learner at the time she has selected. She uses the system to bring up details and previous reflections on related experiences in the past: the system gives her details of two previous presentations she gave [K, L], including her own analysis of the skills used and the confidence in them at the time, and the written feedback she received from her tutor and peers after those presentations. It is then very easy to find relevant guidelines and study material on presentation skills. Guidelines and principles alongside the recollected experience help her to arrive at her own understanding and conceptualisation of what constitutes effective use of audio-visual support in a presentation, and why. She re-assesses her level against the guidelines, and revises her judgement of where she is in terms of presentation skill [N].

The learner consults her tutor, who agrees that this is an appropriate issue to focus on at this stage [P], given her imminent presentation task, and a goal is agreed to improve her presentation style [Q]; analysing the presentations she has watched and made, she decides on a detailed focus on better quality and use of audio-visual aids. The PDP system contains a number of templates for planning presentations, and she uses one she has not tried before [R]. She completes a new plan for her forthcoming presentation [S] referring back to the two presentations she attended recently, analysing them specifically in terms of the quality and use made of audio-visual aids. She also chooses a short practice task to complete with the help of her peer mentor (or "critical friend"). This is recorded in her appointments system as well as that of her friend.

The PDP system reminds them about the exercise [V], and afterwards offers further opportunity for reflection [C, D, G, H]. The presentation itself goes well [Z]. The learner appreciates the positive reactions from her work colleagues [A, D] and records this in her log [H]. Because the individual action was anticipated, the recollection that goes on as an integral part of recording [L] is sufficient, without needing any further recollection, for her to know that she can now use audio-visual aids better in presentations [N]. Privately, she knows that whenever another presentation comes up, she will be able to go through the same process with more confidence.

2 Web services for reflective learning: future possibilities

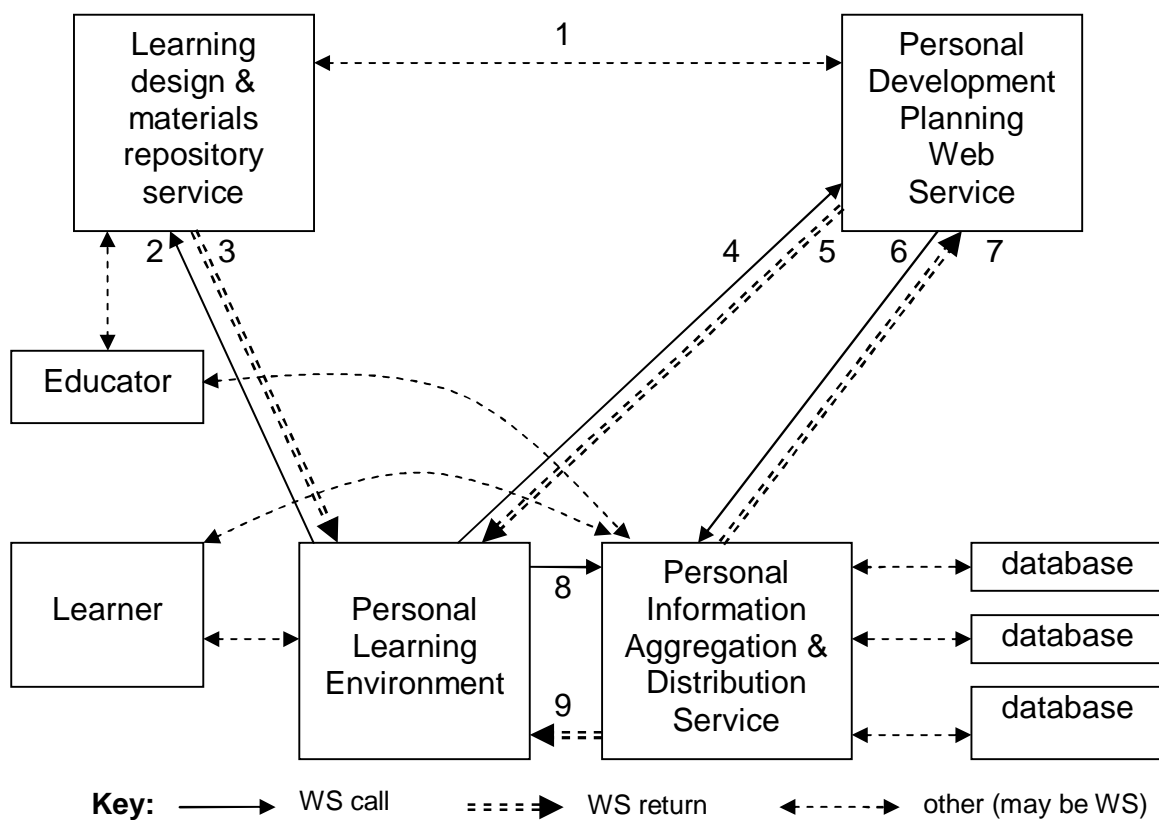


Figure 1: Components and information flows in the envisaged architecture

2.1 The systems involved in this discussion

To understand the nature of the proposed system of web services for reflective learning, the individual components need explanation, as most of these have not yet been implemented. The modular nature of the architecture will mean that the individual parts can be substituted relatively easily, just as long as the calls are based on common WSDL definitions using common XML standards for the return of information.

A central feature of this architecture is the separation of the PDP web service (PDPWS) from the Personal Information Aggregation & Distribution Service (PIADS). This is greatly motivated by the recognition that the information dealt with in PDP largely overlaps with the information useful to present to other people for various purposes. Traditionally, a selection of this information has formed the basis of a CV, and this functionality is expected increasingly to be taken over by electronic portfolio (e-portfolio) systems⁴, able to display chosen items to people given specific permission. In short, personal data such as this will be used by many more systems than those performing just PDP.

We are assuming that the web services described here will be accessed through some sort of Learning Management System (LMS) – we consider two cases a Virtual Learning Environment (VLE) and a Personal Learning Environment (PLE). We feel that the concept is well understood but will present an outline of the key concepts of a PLE.

2.1.1 Personal Learning Environment (PLE)

Much has been written about the concept of the Personal Learning Environment. We are clear that for the most effective functioning of a PLE, a service to store personal information about learning and related matters (such as a PIADS, introduced below) would be very helpful. A PIADS service would also provide the basis of e-portfolio functionality, and could integrate with tools to manage that functionality (including managing the presentation of records to third parties) which could be implemented in a PLE. These tools are a fundamental part of PIADS since access may be conferred by relationship. Such relationships can be recorded within or between different learner's PIADS; access to other people's PIADS could be granted on the basis of inter-PIADS relationships

⁴ The following is a definition of e-portfolios established by the UC Berkeley Leadership Development Program (LDP, <http://hrweb.berkeley.edu/ldp/ldp.htm>):

An e-portfolio is a highly personalized, customizable, web-based information management system, which allows students to demonstrate individual and collaborative growth, achievement, and learning over time.

The key rationale among higher education institutions, then, for implementing e-portfolios for students has primarily been to (1) give students the context in which to reflect upon their social and academic experiences, and to (2) improve upon current career and resume planning. To fulfil these purposes, and others, e-portfolios function in what may be seen as five basic ways, which together work toward placing the student at the centre of both their learning and development. These five functions of e-portfolios make clear their more practical uses and benefits for students. They are:

1. storage
2. information management
3. connections
4. communication
5. development

For more information see: <http://istpub.berkeley.edu:4201/bcc/Spring2004/eportfolio.html>

Thus, one function of a PLE might be to provide a basic browsing and editing interface to the learner's PIADS. Additionally, the personal preferences of the learner (as, e.g. IMS AccessForAll [ACCLIP], see <http://www.imsproject.org/accessibility/index.cfm>) could be stored in the PIADS, and retrieving these would enable the PLE to configure the user interface to suit the learner's preferences and needs. Another useful piece of functionality would be for the PLE to act as the front-end to discovery services. Centrally to the current discussion, a PLE could

- provide an environment for the user to undertake supported e-learning without being on-line;
- manage the registrations with the different Web Services for learning and PDP (registration details could be stored in PIADS);
- manage the actual presentation of information to the user, whether graphical, auditory, tactile, etc.;
- manage accessibility information (e.g. ACCLIP, stored in PIADS) which constrains the above;
- mediate dialogue on presentational styles;
- issue Web Services calls to PDP and learning repository systems;
- play learning materials represented in Learning Design⁵;
- display related information-gathering forms, manage any data submitted and return appropriate structures to relevant servers;
- act as a focal point for the initiation of authentication and authorisation.

If the PLE is going to be able to deal properly with LD that is more than just one form, it would be useful to have some means of storing intermediate results and holding the information for use on other forms, for later submission.

2.1.2 Personal Information Aggregation & Distribution Service (PIADS)

There is no reason why all of anyone's personal information should be held on one server, and many reasons why it should not be. Indeed, the current draft IMS e-portfolio standard (<http://www.imsproject.org/ep/>), section 2.1.3.2 (http://www.imsproject.org/ep/epv1p0pd/imsep_bestv1p0pd.html#1662411) explicitly makes mention of the distributed nature of an e-portfolio store. Reasons include ownership of the information and security. Already, for instance, at an educational institution, some information is held on servers owned by the institution, while other information is counted as privately owned, and can for example be held on a separate PDP system. However, when the

⁵ The IMS Learning Design specification supports the use of a wide range of pedagogies in online learning. Rather than attempting to capture the specifics of many pedagogies, it does this by providing a generic and flexible language. This language is designed to enable many different pedagogies to be expressed. The approach has the advantage over alternatives in that only one set of learning design and runtime tools then need to be implemented in order to support the desired wide range of pedagogies. The language was originally developed at the Open University of the Netherlands (OUNL), after extensive examination and comparison of a wide range of pedagogical approaches and their associated learning activities, and several iterations of the developing language to obtain a good balance between generality and pedagogic expressiveness.

For more information see: <http://imglobal.org/learningdesign/index.cfm>

person wants to access their information, and control other people's access to it, it is important that they can have access to all of its parts, in an easily-understood structure.

Data being what it is, the difference between possession of the data, (i.e. storage in one place,) and ability to access it all from one place, is vanishing small. The important thing is to create the point of access within a business model that makes high quality secure authentication affordable.

Deciding on a main location for every particular piece of personal information should reduce the likelihood of unnecessary duplication. Duplication may still be wanted either for backup, or because systems may want to work independently without an effective immediate data connection. The rules governing backup, synchronisation or caching could also be held in the learner's PIADS, and managed through a PLE or otherwise.

When personal information is created or changed, the learner's PIADS should also be able to direct the information to its proper destination(s) for storage. It should also be pointed out that the underlying learner data may be represented in any format (not necessarily XML). One would expect that the data is transported in 'standards conforming' fashion (XML) but the standard is not necessarily based on an IMS specification. IMS LIP data format will be used, but it is also possible for the PDP system to require access to IMS Enterprise data, HR-XML data (<http://www.hr-xml.org/channels/home.htm>), vCard (<http://www.imc.org/pdi/>) and even 'the specification formerly known as PAPI' (<http://edutool.com/papi/> and <http://jtc1sc36.org/doc/36N0479.ppt> [comparison of PAPI and IMS LIP]).

Having such a service would be a central support for the implementation of e-portfolio systems. A crucial point to bear in mind here is that an e-portfolio not only acts as a personal record and repository, but as a system to expose selected parts of that information to others, either publicly, or by class of person, or by individual. A typical e-portfolio owner may want to keep many personal reflections private; may want to allow, say, all people in a peer group to see contact details and information about group-oriented work; and to show particular CVs to specific people in companies to which the owner is applying for employment.

There should be just one operating PIADS for each learner, which would:

- act as a virtual database for all learner profile information and files, for everyone allowed;
- maintain identifiers for all logically distinct pieces of information – for XML documents, this involves inserting XML id= attributes to all nodes so that a common method of referral can be used regardless of which IMS (or other) specification is being used;
- accept calls for information from PLE or PDP services;
- allow the owner to manage authorisation to view all separate parts of the information held.

There are services either proposed or under development which aim to do related things – for example,

- ioNodes (<http://www.phosphorix.co.uk/around/ioNode/>) at an institutional level – but as far as we know, this does not yet extend to the learner managing their own personal information, which is the distinguishing feature of a PIADS.
- GuanXi (<http://www.guanxi.uhi.ac.uk>) which is developing an open source configurable attribute agent (or so-called mega-origin) for use with Shibboleth⁶. This agent will act like a meta-directory in that it collates a learner's personal information from an array of attribute stores such as Student Record Systems, LDAP and VLEs. The learner (and indeed Shibboleth) will view their attribute store as a single entity rather than the distributed network that it really is.
- Virtual Home (VH) (http://www.edentity.co.uk/tr_overview.html). This can be thought of as a point of presence in the network space from which an individual can give permission for the sharing of personal information between third parties. Virtual Home has a close relationship with PIADS and the ideas behind it are quite well developed. It is felt that there could be a good working relationship here.

In a PIADS, the personal information needs to be structured by the user in order to be browsed and managed effectively. This is not conceptually difficult to imagine, as it is what happens already in a normal file store. A hierarchical storage system is most familiar, and has the built-in property of being able to inherit permission settings, but this is not the only conceivable way of structuring the one's information. However, it is by no means given that the structure of an interoperability standard will be the most natural and intuitive way for anyone to arrange their personal information.

In principle, there need to be two mappings involving this structure for the PIADS concept to work. Firstly, a mapping between the personal view and the storage view: for any piece of information in the personal structure, there has to be a rule for determining where it is actually held (potentially more than one place as well) and what it is called in that place. To keep track of this mapping, the PIADS has to maintain a table of correspondences between the PIADS ID of any piece of information, the system it is stored in, and the ID on that system.

Secondly, the mapping between the personal view and the interoperability standards view: assuming that a PDP service conforms to an interoperability standard, this mapping will allow the user's structure to be mapped onto that of the PDP service. Both these mappings have to be feasible: this means that the concepts at the lowest level of granularity must not overlap.

Given that these mappings are sensible (whatever that means mathematically) they should be able to be represented or even executed by something like XSLT. Hopefully, more than that: if the data stores are mapped fully against the interoperability specs, then one mapping from the user should be enough: just compose the two mappings (as it were on two sides of the triangle) to get the third. Major problems may occur if the mappings are not composable in that way: a situation might arise where three such mappings are incompatible.

Thus it makes sense for an interoperability spec to be the main (or primary) target for mapping from the user's structure. The user defines the relationship between their own conceptual structure and that of the spec; the data stores do the same; and the storage mapping should be able to be worked out. A good interoperability spec needs to be able to represent

⁶ See <http://sourceforge.net/projects/siva/>

granularity right down to the lowest level likely to be operational in the data stores. Thus the granularity of the interoperability spec can serve as the main guide to the ideal level of detail of storage of the information. To take a short cut, one may also imagine that the learner is able to represent their own information in terms of such a spec.

WS calls to the PIADS need to identify who the subject of the information is, and who is the person or authority making the call. There is a need to authenticate these identities. (This is an integral function of PIADS / Virtual Home - since the authentication requires matching of supplied data against known data, which can be stored internally. Note that this begs the question of what is meant by authentication - and requires the acceptance of the idea of many identities and the difference between single domain and cross-domain authentication. The idea of trusted relationships between two instances makes the system very powerful but also puts a lot of pressure on managing safe authentication.)

Our inclination would be to place PIADS in the Common Services section of the JISC e-learning frameworks building blocks structure <http://www.cetis.ac.uk:8080/frameworks> and call it something like "personal information management"; this seems to fit with the term "content management". We are unclear as to whether this is actually the same thing as the "filing" component, we suspect not.

2.1.3 PDP web service (PDPWS)

The role of the PDPWS is to provide structure and guidance to learners for PDP-related activities – that includes any activity which is to do with transferable skills, and particularly with learning about learning (one aspect of metacognition). Having PDP as a web service enables us to reclaim the established definition of PDP as “a structured and supported process undertaken by an individual to reflect upon their own learning, performance and / or achievement and to plan for their personal, educational and career development”, which says nothing about interaction or storage technologies or systems. With the e-portfolio functionality – to do with storage and management of the information – handled by the PIADS, the role of the PDP service is just to provide the structuring to the process, tailored to the individual. In order to do this, the PDPWS would perform these kinds of function:

- maintain lists of authorised learners and their permissions;
- hold learners’ history of interaction with the PDPWS;
- hold PDPWS-specific information about learners, e.g. PDP programme signed up to;
- hold a detailed “map” of all possible “positions” of a learner’s PDP progress (this “map” may include a skills/competency framework);
- hold designed procedures for determining what to return for a user in a position (these procedures may constitute “paths” through the system);
- offer an interface for those requesting registration;
- discover the location of learners’ PIADS – this will be done by passing the location of the PIADS as a parameter to the WS call.

Thus, designers of the PDPWS need to design:

- the “map” defining the possible locations in their system’s version of “PDP space”;
- the process for each node in the map, dependent on any relevant attributes of the learner.

This rather abstract definition can be illustrated by comparing it with the LUSID web-based PDP system (<http://lusid.liv.ac.uk/>). The current “where you are in LUSID” is given by the page and the query string (which includes record operated on) and the data store and history. This is quite close in feel, though different in approach.

2.1.4 Repository of learning materials and designs

It has been often remarked that VLEs (“virtual learning environments”) are often initially used as repositories, or content management systems, for learning materials. Here, this is designed to be the basis of the role of this repository. Other functionality that has been provided by VLEs could also be served either by the same system, or in different ways by various other services.

Effectively, the repository would be like a VLE without the front-end, since the front-end is managed by PLE. It shares many of the features and characteristics of the PDP Web Service, but is in general not tightly bound to the characteristics and history of individual learners.

2.1.5 The databases

The principle in this architecture is that much information about any learner will be held separately on various databases – this is already the case in many places; and therefore this architecture is much more realistic than architectures which envisage the centralisation of all learner information. Obvious current databases include student record systems (SRS) in educational institutions, current PDP systems, company HR systems, etc. Another possibility is to treat any standard information services similarly. For instance, if the information specified in the EDUCAUSE/Internet2 eduPerson could be held in a separate service, and access to it could be controlled, along with all other personal information, by the PIADS. This is an obvious way to achieve consistency in privacy, which might otherwise be hard to control.

2.1.6 A note about security, authentication and authorisation

This paper does not deal in great detail with any authentication or authorisation issues: they are regarded as beyond the scope of this document. We will, however, present a brief guide to the options which can be considered.

There are basic issues of authentication with both services, is the client allowed to access the web service and, if so, is the individual learner allowed to access the services that are provided? In the case of PIADS, we must also consider the authorisation to view / edit learner data – a third party may want to view data, has the owner given permission for this to happen?

Initially, individual authentication will effectively be performed by the host Learning Management System (LMS) system, (i.e., the web service client); this may include a single sign on solution such as WebAuth. In time it may become practical for an institution to outsource the authentication decision to PIADS and thereby share the cost of high quality tokens etc with other service providers.

The host LMS may also support the SAML-based Shibboleth (<http://shibboleth.internet2.edu/>) framework or a similar technology Microsoft's .NET Passport scheme. Once the learner has logged in then they should be able to use all the web services that their LMS is authorised to access. In time, it may also make sense for PIADS to integrate Shibboleth (or equivalent) functionality: education providers will write permission attributes for specific on-line resources to an individual's PIADS.

The next stage is to ensure that data can be securely transported. There are a number of ways of ensuring transmission security; the simplest will be to use HTTPS as a secure transport mechanism. The server (WS) will identify itself to the LMS via a digital certificate which also contains the server's public key; this key will be used by both parties to initiate symmetric encryption. The client will then send its own certificate (and public key) to identify itself to the server. It is understood that the WS will have a look-up table of 'registered clients' that can be checked for authorisation. The whole SOAP message will be encrypted for transport and will be unencrypted at the other end ensuring total privacy. This message is quite simple and works well with small amounts of data. (The encryption of the whole message will incur a sizable overhead but if the amount of data being transferred is small then this will not impact too greatly.)

A more scalable solution will be to use WS-Security (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss). This reduces the amount of processing by only encrypting *sensitive* information. In this case the client will supply a signed SOAP message that will be checked by the target WS for authorisation to use the service. The client will also want to know that it has contacted the real target service (and not some impostor) so a certificate must also be sent back to complete the handshake. Obviously, this will alleviate the need for HTTPS.

The PDP WS is only used by the learner, their credentials are supplied as request parameters to the service; the PIADS on the other hand can be used by all and sundry. The first thing either web service must do is check that the user credentials passed in the service call authorise the applicant to actually use the system; it will use some sort of internal look-up mechanism for this.

In the case of PIADS, once the call has been securely made and (information about) the desired records assembled, the system must decide whether the requestor has authorisation to view the learner's records. PIADS will use the learner's visibility attributes (typically created within an e-portfolio) and check them alongside the credentials presented by the requesting authority to ascertain viewing and editing rights.

2.1.6.1 Implementation in a Shibboleth Environment.

The web service calls by the client will include attributes that must be retrieved from an attribute authority (AA) at the learner's home institution. These attributes could be managed by Shibboleth, indeed, they will be supplied by the learner's attribute authority.

There are 3 recognised ways of achieving this.

1. The PLE authenticates the learner and receives an encoded Shibboleth ID – a handle into the AA. The PLE then contacts the AA and gets the relevant PDPWS user id,

- location of PIADS, group membership and so on. These attributes are placed in SOAP message body as parameters to the service calls and dispatched to the PDPWS.
2. The PLE authenticates the learner and puts the Shibboleth handle into a SOAP message as an attribute to Shibboleth-aware service call which it sends. The PDPWS then retrieves the learner's attributes from an AA. This solution requires that the target web service is Shibboleth-aware.
 3. The PLE calls a new intermediate proxy and supplies the learner's Shibboleth handle; the proxy gets the required attributes from the AA, rewrites the SOAP message to include recently fetched attributes, and passes to the PDPWS.

Rationale for choice

Web Services are loosely coupled components where the client side is the front-end (used for integration) and the server side holds the real business logic. In this context, the server side should be as focused on the task it is supposed to do, and it should be as independent of system features (e.g. authentication) as possible.

So, in most of the situations (if not all) they should not know anything about Shibboleth. If you need to have the notion of a user ID at the server side, this should be passed by the client side of the WS.

There is no notion of user ID (e.g. username, uid, unique learner id) in Shibboleth: that is in order to preserve user anonymity. However, in an e-learning context using Shibboleth, the user ID might be an attribute that a user is happy to release, mainly because there will be many targets requiring it in order to authorise access. The learner will have to take care to manage their anonymity with e-learning systems.

Note that in this approach Shibboleth is integrated in a non-intrusive way. The PLE doesn't have to deal directly with the AA as this is deferred to a proxy. More than one learning environment will wish to use a PIADS service, each can use the proxy and thereby decouple themselves from authorisation tasks. Other, alternatives might be more adequate according to the system requirements.

So, in the above list, 3 is the best method. The proxy can be told to decode the handle or not dependent on the type of data being sent. (There will be an attribute in the SOAP header telling proxy to contact Shibboleth or just to pass the message on without rewriting.) It may be that a non-Shibboleth system is being used to send the data. Also, there is no need to integrate the PDP and PIADS WS with Shibboleth as it is all handled by proxy. The proxy can rewrite SOAP message to include ALL attributes required by PDP service (eg, username, vCard info, host institution, PIADS data store location, language, accessibility requirements and so on).

If approach 3 is implemented then the PDPWS and PIADS do not have to be integrated with Shibboleth which is a big advantage, but it does mean that we have to pass in all learner attributes via the WS call. The learner's PIADS will allow much information to be fetched, name, address etc. In this sense the PIADS acts as personal AA but is different to a Shibboleth AA in the sense that it can be accessed by a third party subject to the access permissions set by the learner.

2.1.7 The relationship between PIADS and a Shibboleth Attribute Authority

A Shibboleth target (SHAR) will ask an attribute authority for details about a certain user. This AA, also known as a Shibboleth Origin can be configured to draw its data for one or more data sources; typical sources would be an institutional LDAP, a student records system or maybe a VLE. This is a read-only process and is tightly monitored by the 'owner' of the data (here 'owner' is taken to be the person to which the data refers).

PIADS in some ways behaves like an AA except:

1. it is not tied to Shibboleth
2. it allows third parties to access the data
3. access is controlled by properties embedded within the data
4. it allows both read and write access
5. it is not owned by the learner's education provider but by the learner themselves

We envisage that PIADS will be one of the data sources that the attribute interrogates when building up the list of possible attributes. Indeed, in many situations, it may be the only data source as it may provide access to all underlying data.

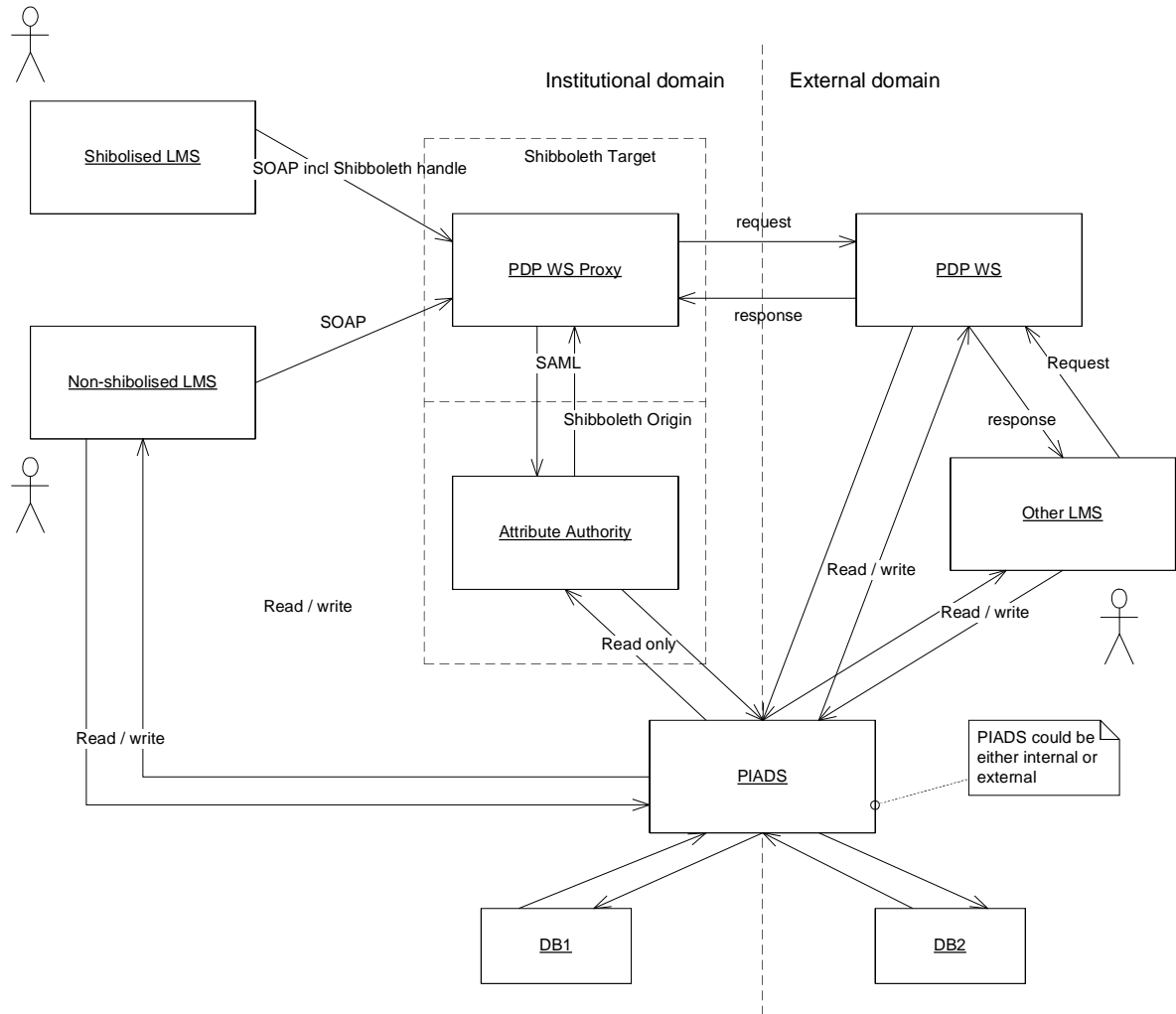
There is an interesting situation here; the PIADS data (attributes) can only be accessed if

1. authority has been granted to use the PIADS service
2. permissions on the data are such that the enquirer is allowed access.

Shibboleth's AA effectively allows only the subject of the data access. The subject must then 'rubber stamp' the release of attributes which are passed to the target application. These applications are not registered with PIADS and do not even know its location. (Note also that the subject of the data is not necessarily the owner – the data may be owned by the subject's institution.)

[As a side note, it should be pointed out that anonymity is one of the cornerstones of Shibboleth, however, in an e-learning context there is very little that can be done without releasing an identity attribute.]

The following diagram is an attempt to represent the position of PIADS within a Shibboleth-aware environment. It shows that the whole system would hang together if the decision were made NOT to pursue Shibboleth for authorisation.



The above diagram tries to show the relationship between a Shibboleth-enabled (Shibbolised) WS proxy, various Learning Management Systems (LMSs) and the Personal Information Aggregation / Distribution Service (PIADS) data store. PIADS can be accessed by any web service-enabled system with sufficient authority to do so.

The use of an intelligent proxy will allow the target web service to be used with both Shibbolised and regular (non-Shibboleth-enabled) LMSs.

The SOAP message which is passed to the proxy may contain a 'rewrite' switch in its header. If the switch is set, then the PDP WS proxy will act as a Shibboleth target and extract the Shibboleth handle, the relevant attributes are then retrieved from the learner's attribute store by using the Shibboleth origin's Attribute Authority (AA). The SOAP message will be rewritten to include these recently fetched values; the recomposed SOAP message will be sent to the final PDP WS. The AA may use PIADS as one of its attribute stores; obviously the AA may only read from PIADS whereas other applications may read and write.

The WS proxy will simply forward any messages that do not require rewriting.

No rewriting is necessary when the response is received.

An architecture such as the above allows the full benefits of Shibboleth to be enjoyed without requiring the destination PDP WS or PIADS to be tied to Shibboleth; the WS can also be used directly by other clients.

PIADS may be hosted either by an institution or an external service provider.

2.2 First use case: a small PDP activity embedded in a short course

[Numbers in square brackets refer to the numbered information flows in Figure 1.]

Say an education provider wants to deliver a short course in Evolutionary Psychology. The main components of this short course might be, say:

- read some literature;
- engage in a group discussion;
- write an extended essay.

We imagine that the level of essay-writing ability in the possible course attendees is not certain, so the educational activity designer wants to add in a brief component on essay planning, which is considered to be in the domain of PDP. The designer could use a discovery service to locate suitable PDP web services. Using a learning design tool, the designer could specify an appropriate activity to help with planning an essay – there may be many of these on offer; meta data should be provided about the activities so that the designer chooses tasks which match learner expectations.

Designers may want to choose between different options for how to specify the one considered best [1]. Firstly, they could simply specify the characteristics of the learners, and choose something at their level: this could be called “anonymous” PDPWS use. Also anonymously, designers could arrange for information about each learner individually to be passed to the PDP system, so that a principled choice could be made of a suitable level of PDP activity. Secondly, they could arrange for a PDP activity that had no immediate relationship to other possible PDP records, but still allow the PDPWS to interrogate appropriate PIADS systems about the characteristics of learners, so that an appropriate level of activity can be served, for classes where there may be mixed levels of personal development: this could be called a “dislocated” PDP activity. Thirdly, there could be “situated” PDP activities, where, for example the designer planned for the learners to review previous essays that the learners had written in the past. Equally, the learners themselves could select an essay planning service from a list, or even take responsibility for searching for a suitable PDP service.

We can envisage the flows of information at “learn-time” in the simpler case where the designer has chosen the essay-planning activity. The learner is registered with the learning provider which is hosting the short course. As part of signing up for the course [2], the learner allows the course outline schedule to be passed to their PIADS, where it will be available both to their personal calendar system and also to the PDPWS for action planning purposes, as well as later reflection.

The overall Learning Design (LD) instance for the short course is retrieved from the Repository server to the learner’s PLE [3] using a ‘search’ service from the common services layer of the JISC e-learning framework (ELF). (A PDP system would be expected to provide this web service to allow searching.)

Initially, the LD guides the learner to read one or more items from a reading list, and to connect with a resource management service to help with that – this is not part of the service to be considered here. Also included in the LD may be some exercises to help and to assess comprehension of the reading material, and also the kind of typically PDP activity that involved metacognition – awareness of the learner’s own learning processes and characteristics. After that, there is a group discussion activity, which is scheduled in the LD, but again delivered through a separate service.

When it is time for the essay planning, the LD provided to the PLE directs the learner to the PDPWS [4]. The PDPWS may interrogate the learner’s PIADS to discover the level of development of the learner both in essay planning and meta-cognitive skills [6, 7]. This allows it to customise the language used, and the amount of support given to the learner – very important if the service is to avoid the complementary dangers of over-complexity leading to the learner giving up, and over-simplicity leading to the learner rating the service as unhelpful or even stupid.

The LD for the essay planning activity is packaged along with the information needed to specify the essay-planning activity itself [5], and returned to the learner’s PLE.

The LD then returns to the rest of the course, including the writing of the essay. The essay is sent to the PIADS which stores it in the learner’s e-portfolio [8], and if appropriate sends it also to the education provider for marking. This may not be necessary if the education provider is able to access the learner’s e-portfolio – all that would be necessary is the essay is made readable by the learner for anyone marking or commenting on it.

2.2.1 Web services required for the first use case

Two distinct types of service are needed in this kind of scenario: firstly provision of learning materials and services (provided by the repository and the PDPWS), and secondly storage and management of information (provided by or through the PIADS). They differ in several ways. The learning materials have to be authored, and are liable to require digital rights management of some kind. Learning services may include provision of tutors or other paid advisors or experts as part of the learning process. On the other hand, storage and management of information has neither of these requirements, but it does need to be done independently in principle, because there are other uses of that same information.

We can also distinguish between the actual databases providing storage services, which may be distributed; and the aggregation and distribution service acting as a virtual database, which we have here called the PIADS.

The PDPWS call should return IMS LD containing either embedded WS call-backs to the PDP system to fetch the data entry pages, or the data and pages themselves. Ultimately these pages should comply with accepted standards and be expressed as xhtml, XForms plus an IMS LIP model (or whatever). The model combined with the learner’s input data should be sent to a PIADS WS for storage; however, as XForms do not yet support embedded WS calls, this cannot currently be done. In the future, we envisage the information from the XForm would be sent to the PIADS, which then distributes it out to the appropriate places.

If an essay plan has actually been constructed as part of the essay planning activity, this may be returned to the PLE to be used in the actual writing of the essay. Following this through, the LIP for the essay plan will naturally occur in the returned XForm model.

It should be highlighted here that one of the key concepts of PLEs are that they are NOT ‘always online’; moreover, they are specifically designed to be usable where no internet access is possible. This means that there needs to be an approach where the IMS LD that is downloaded also contains the learner’s data; it is up to the PLE to ensure that the integrity is preserved. The PLE will have to merge the learner’s data with the XForm using a just-in-time methodology. At the end of the LD, the PLE will hold updated records which must be communicated back to the learners PIADS as soon as the PLE has internet access. It is envisaged that an agent will manage the storage behind the scenes.

The PDPWS must also be usable in contexts other than PLEs, for example, within traditional ‘always on-line’ VLEs. In order that the burden of data management placed on the PLE does not also have to be placed on a VLE, it would seem prudent to offer an alternative online solution. In this case, the LD does NOT contain the XForm plus user data; the form is fetched from the host PDP system on-the-fly and then the data is immediately stored when the form has been completed. This is the more traditional way of doing things. In this case, the initial WS call to the PDPWS to return an input form could be regarded as the learner logging into a PDP system (as they must have authenticated to do this). This means that it is perfectly possible for a PDP system, e.g. LUSID, to return proprietary servlets-based HTML data input pages. These pages can work in exactly the same way as they currently do: authentication is provided by the addition of a session id ‘hidden field’ (which would be generated and attached when the WS call is made to fetch the page); the data input fields could be identified using LUSID’s naming schema; and the action of the form is to invoke LUSID (using a HTTPS POST operation) such that it will store all the data entered on the page (just as it would do now if a submit button is pressed). In this case, LUSID will act as a PIADS proxy and it will extract the data from the form, compose it into a LIP record which is then passed onto an appropriate PIADS service.

There are potentially many services that might want to use the PIADS, so it would be better to allow the PIADS to take information directly from any service, rather than designing it to go through a proprietary proxy – the implementation there would only have to be duplicated.

We also note that the requirements for XForms 1.1 include the ability to use SOAP as a submission method. This suggests that by using XForms in the future, we could if desired restore the full SOAP-based Web Services approach to all communications.

2.3 Second use case: learner-directed PDP

As an alternative kind of example, we can look at the scenario envisaged for reflective learning and personal theory building. Learners currently use PDP systems directly for making learning logs, planning, and finding skill development resources. In this future scenario, the PLE would be acting primarily as a front-end to the PDPWS and the PIADS.

Presenting a “chunk” of PDP in this scenario is essentially similar to the previous one. What is different is that here, the learner will not necessarily be directed by an educational designer

to a particular activity. In learner-directed PDP, the learner will want to decide what it is time to do now. The reflective personal theory building scenario contains several possible instances of PDP-type activities (or tools):

- a learning log;
- a reflective / analytical tool for analysing experiences (including log entries);
- a scheduler / work planning tool which can issue reminders;
- searching for previous similar experience and presenting it;
- potentially mediating consultation with tutor / mentor;
- setting goals and creating action plans for their fulfilment;
- providing signposts to developmental resources;
- skills profiling tools.

Most of these processes involve interaction with the learner's personal records. Whenever personal information is needed, it would make sense to go through a staged process. Firstly, check if there are any up-to-date records stored locally. This could have been arranged through synchronisation, caching or mirroring. If the information is not held locally, or if the currency of the local records is in doubt, request the information from the PIADS. If the PIADS has not got it, or cannot release it, ask the learner directly, or manage without the information.

One very general question to resolve in this learner-directed interaction is how the learner starts off the interaction with the PDP system to get the appropriate activity down from the PDP server, ready to "fill in" and send back.

At present, in LUSID, this can be done in two ways. The learner either follows the path, or chooses from the menu, provided by the system, or looks at the records stored in the system and from that decides what to do. The fact that different pathways suit different learners is reflected in the fact that LUSID "navigational" pages can be constructed, just like all LUSID pages, in a way customised for any particular group of learners. A fully-self-motivated learner may just want to see the overall menu in its entirety, and choose from that. In addition, the system may be able to provide suggestions about what to do: take up where it was last left off; fill in something that is clearly missing; deal with overdue items; or perhaps some more subtle and sophisticated recommendation based on the current situation and profile of the learner.

In many cases of learner-directed PDP, there is likely to be a multi-stage interaction dialogue with the learner before a "content" chunk of PDP is ready to be served. The choices to be made in the course of this interaction may be ones for the learner alone, or they may invite a decision process involving other people – mentors or parents for example. The question is then, how is this multi-stage interaction to be carried out with the PDPWS?

In the background of this discussion lies the general assumption that learners will be registered for particular PDP services, in the same way that they may at present be registered for doing PDP with generally one body or institution. However the concept of a PDPWS invites the idea that a learner can search the web for suitable PDP services. This is possible when the service offered has practically no cost to the provider, but that would imply that the PDPWS is doing the "structuring" but little or no "supporting". Much more likely, and more plausible, is that substantial PDP services will be offered to subscribers: in an academic institution, fee-

payers; in the work context, employees; or to paying members of trade or professional associations or unions.

There are two issues which may come to the fore in discussing the interaction of a learner with a particular PDP service: the starting point, and the finishing point. The starting point is to answer the question, what is the initial service call that is made to the PDPWS? Is it simply something like “what PDP can you offer me”, or something more detailed? The finishing point is, at what level of granularity is the navigation (or decision about what to do) over, and some substantive PDP activity ready to start?

However, in the case of the starting point question, our intention is to make the PDPWS as accessible as possible at any level, to allow entry either at the top level, or at any other specific place (as in the embedded scenario). So how does a learner search a particular PDPWS for the PDP desired? There are three ways we envisage:

1. using a keyword search;
2. drilling down through a hierarchical structure to the desired functionality;
3. choosing from a full list of services, organised as nested lists.

To make this effective, the PDPWS has to be able to return information about each node offered (which could be regarded as metadata) such as a title, a description, and a means of selecting the chosen option: envisaging a button here is probably more helpful than a link, as a link would imply a purely HTTP-based mechanism. The metadata required is a subset of many metadata specifications, such as Dublin Core (<http://dublincore.org/>) and LOM (<http://ltsc.ieee.org/wg12/>). Also of interest is the OWL Web Ontology Language (<http://www.w3.org/TR/owl-semantics/>)

A second point is that the leaf nodes are also arbitrary. It may be possible in many circumstances to package up several pages of PDP activity (including forms such as XForms) into a larger piece of LD, or to have them delivered in separate packages. Putting these two points together, it appears that two variants of just one service call will cover the whole range of PDP web service at learn-time.

Rather than detail the web services needed at this point, we are in a position to set out all the web service calls, and then return to working through particular cases.

2.4 Web Services calls and returns

Please note that all services return a status code in addition to the responses enumerated here.

[Numbers in square brackets refer to the numbered flows in Figure 1 above.]

2.4.1 PIADS calls

PIADS will manage the storage of records; it only makes sense to require it to implement, create, store and delete operations, behind the scenes the service will issue the relevant CRUD command to the target back-end database. The authentication of messages will be based upon a digital signature found in the SOAP header; if necessary, the message could also be encrypted; it is also understood that HTTPS should be used to transport the message.



We have not detailed how PIADS manages its aggregation, distribution and synchronisation rules (part of [8, 9]).

2.4.1.1 PIADS “read” [6, 8]

This will fetch data related to the stated learner (subject to view ability restrictions) from PIADS. It is implied that the PIADS will contact all the learner’s data stores and (conceptually) collate all records into one document. This document will be processed in order to only return the specified structures.

2.4.1.1.1 Request Parameter(s)

This is a straightforward call to get information about an individual learner, either by the learner or by a third party authorised by the learner. The PIADS may need to aggregate information from various sources. Rules about which information is held in what external databases are not passed in the call, but held by the PIADS. One should bear in mind that this call should ideally also serve for e-portfolio use, where the person requesting the information may not be the learner. The learner may want to allow access from some specific systems for some purposes: thus the PIADS needs to know the identity of the requesting system as well.

Parameters:

- Learner ID. Use IMS Common Services UserID data type for this argument, (see above). This should not contain the password but must contain the username of the learner whose records are to be accessed.
- ID of requesting authority. Again use IMS Common Services UserID. This time the password *must* be supplied to ensure that the requestor has the right to use the service. This UserID will generally be the details of the learner; however, when a third party is browsing a learner’s records then this UserID refers to the browser NOT of the learner. This is why there are two UserID being passed as arguments.
- Language preference list. Use a sequential list of IMS Common Services ‘language’ objects, see http://www.imsglobal.org/es/esv1p0/imscommon_infov1p0.html#1588786. Note that we will have to house this element in a custom wrapper which allows a sequence of languages to be proffered.
- the namespace and schema location of the specification or application profile in which the information is to be returned; these will be bundled together as one parameter.
- an XPath expression (which can itself be a complex compound) selecting all of the required information in that specification or profile. This is a String.

Note that a service will only support requests for particular specifications or application profiles; any unsupported specification will result in the service returning an unsupported status code.

We use a complex XPath expression rather than a full XSLT, as using an XSLT would turn the web service into a grid service in the sense that the service has to run 3rd party code with all the security problems which this implies.

One can imagine the XPath expression being processed by analogy. Imagine that a learners LIP record is printed out on paper. Each part of the XPath argument represents a section of the potential xml file that can be highlighted, when all the parts of the XPath expression have

been processed, all the highlighted parts of the XML record are dispatched to the PDP system. The rest of the record is not needed by the current request.

2.4.1.1.2 Response Parameter(s) [7, 9]

- XML instance following the appropriate schema.

If the request cannot be implemented, a notice to that effect is returned. (If it can be partially satisfied, a suitable return could be designed either to let the caller know what information is actually available, or to notify the caller of the parts which were not available, along with returning the information which is available.)

In order to return this XML, the PIADS could maintain a set of static XSLT files to transform the internal representation into the appropriate XML format. This can then be filtered with the supplied XPath expression to produce the XML to be output. We do not want to pass an XSLT file as a parameter, as full XSLT would give too much power to the call.

The question of where the permissions and authorisation are checked is an implementation detail.

2.4.1.2 PIADS “store” [8]

This will store the submitted data in stores related to the stated learner (subject to edit-ability restrictions; PIADS is responsible for ensuring that that the data is stored in the correct place.

2.4.1.2.1 Request Parameter(s)

This call is to store information about a learner where it can be retrieved again by the PIADS. The PIADS may distribute any information to the appropriate storage services. (As the XML may differ for different XML specs, a separate call could be used for each different spec: e.g., “storeLIP”, “storeACCLIP”, “storeHRXML”, etc.) The parameters of every call will, however, be the same: the only difference is in the schema which governs the XML document so we feel a generalised approach is best with the type of data to be stored being specified in the WS call. The parameters will be:

- User ID. Use IMS Common Services UserID data type.
- ID of requesting authority (UserID see above).
- the namespace and schema location of the specification or application profile in which the information is to be returned, see above.
- XML document specifying what information is to be deposited, this must preserve the id= attributes generated by PIADS.

The PIADS takes the information submitted, and following rules held internally (managed by the learner) distributes the information to the various proper destinations, with the appropriate calls or formats. If more than one specification is needed, the XML files can be put in separate calls.

The XML document should include identifiers for each appropriate element (e.g. “SourcedId” in LIP), unless they are known to be new ones. Where the identifier is the same as an existing

one, new information will be merged in with the existing. Where no identifier is supplied, new elements will be created.

2.4.1.2.2 *Response Parameter(s) [9]*

- none

PIADS returns acknowledgement if all information is stored as a status code.

It has also been suggested that it returns an XML file of all the information which could not be stored. This might be useful, particularly in cases where the PIADS system had not yet been fully configured, and some information had yet to be allocated to a store. For example, the information could be stored locally, or temporarily in the PLE or somewhere else, though this would of course not be available to the PIADS. If the learner had made a specific decision not to have some particular information stored, it would also be reassuring to receive it back in the knowledge that it had not been stored.

2.4.1.3 PIADS “deleteItem”

This can also be invoked by the learner or a third party; a third party may have supplied a (read-only) testimonial for the learner. There are two approaches that could be taken:

1. only allow one node to be deleted at any one time. The delete call would specify a single element ID to delete. This is a cautious approach.
2. allow many nodes to be deleted, these could be specified using XPath. This would be a very powerful feature with the ability to delete all records at the click of a button!

We will adopt the former. In both cases the concept of a recycle bin may be appropriate so that mistakes could be rectified.

2.4.1.3.1 *Request Parameter(s)*

- User ID, see above.
- ID of requesting authority (normally the learner), see above.
- the namespace and schema location of the specification or application profile in which the information is to be returned, see above.
- ID of element to be deleted, (must use id= attribute of outermost tag block to delete).

This is needed in conjunction with the “store” call, in cases where information is to be destructively replaced. The “deleteItem” call is followed by the “store” call. If there is no “deleteItem” call, all previous information will be retained unless specifically overwritten, where a component element can only have one instance.

2.4.1.3.2 *Response Parameter(s)*

- none

The response can be simply a return of success or failure (status code).

2.4.1.4 PIADS “deleteItems”

As above but deletes multiple items.

2.4.1.4.1 Request Parameter(s)

- User ID, see above.
- ID of requesting authority (normally the learner), see above.
- the namespace and schema location of the specification or application profile in which the information is to be returned, see above.
- List of IDs of elements to be deleted, (must use id= attribute of outermost tag block to delete).

2.4.1.4.2 Response Parameter(s)

- none

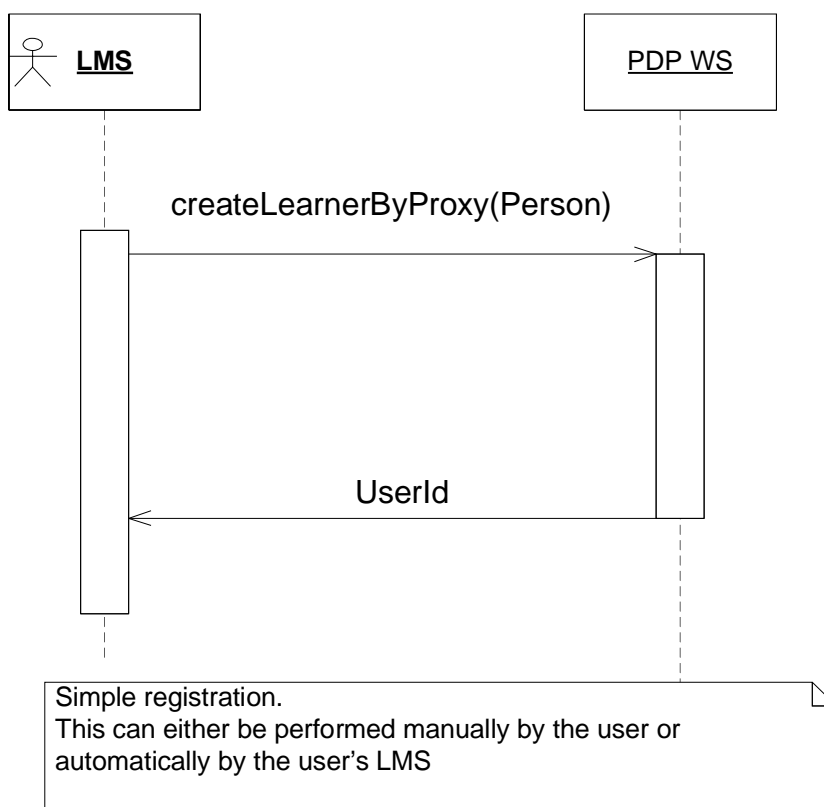
The response can be simply a return of success or failure (status code).

2.4.2 PDPWS calls

The service should include a facility to create and delete and manage accounts on the target PDP system (for example, “createLearner”, “deleteLearner”). There is also a search service for the learning designer, plus services to return specified PDP processes (as IMS Learning Design and xHTML / xForms).

2.4.2.1 PDPWS “createLearnerByProxy”

This will be carried out in a similar fashion to that employed by the Person management facilities of the Enterprise Services model (see <http://www.imsglobal.org/es/index.cfm>). Our process is slightly different as we will be creating a learner account and will need to return account details to the client.



2.4.2.1.1 Request Parameter(s)

This is analogous to the 'createPersonByProxy' service from the IMS Enterprise web service model (see <http://www.imsglobal.org/es/index.cfm>). Parameter:

- Person data object (see IMS Enterprise Services, http://www.imspjroject.org/es/esv1p0/imsperson_infov1p0.html#1531497). (It is anticipated that systems will require different information for a successful registration, for example, LUSID will reject a request if the Person object does not contain an email address; other systems may have different requirements.)

2.4.2.1.2 Response Parameter(s)

- A UserID object containing the username and password, (see IMS Common Services, http://www.imsglobal.org/es/esv1p0/imscommon_infov1p0.html#1589509)

2.4.2.2 PDPWS “createLearner”

This is very similar to createLearnerByProxy except that here the UserId is supplied by the caller. See above.

2.4.2.2.1 Request Parameter(s)

- UserId object (includes username and password). See above.
- Person data object (in the case of LUSID must include email address plus names). See above.

2.4.2.2.2 Response Parameter(s)

- <none>

2.4.2.3 PDPWS “deleteLearner”

This is very similar to createLearner except in reverse!

2.4.2.3.1 Request Parameter(s)

- UserId object (includes username and password). See above..

2.4.2.3.2 Response Parameter(s)

- <none>

2.4.2.4 PDPWS “changeLearnerIdentifier”

This is similar to createLearner except that here an existing (source) UserId is replaced by the (target) UserId supplied by the caller..

2.4.2.4.1 Request Parameter(s)

- Source UserId object (includes username and password).
- Target UserId.

2.4.2.4.2 Response Parameter(s)

- <none>

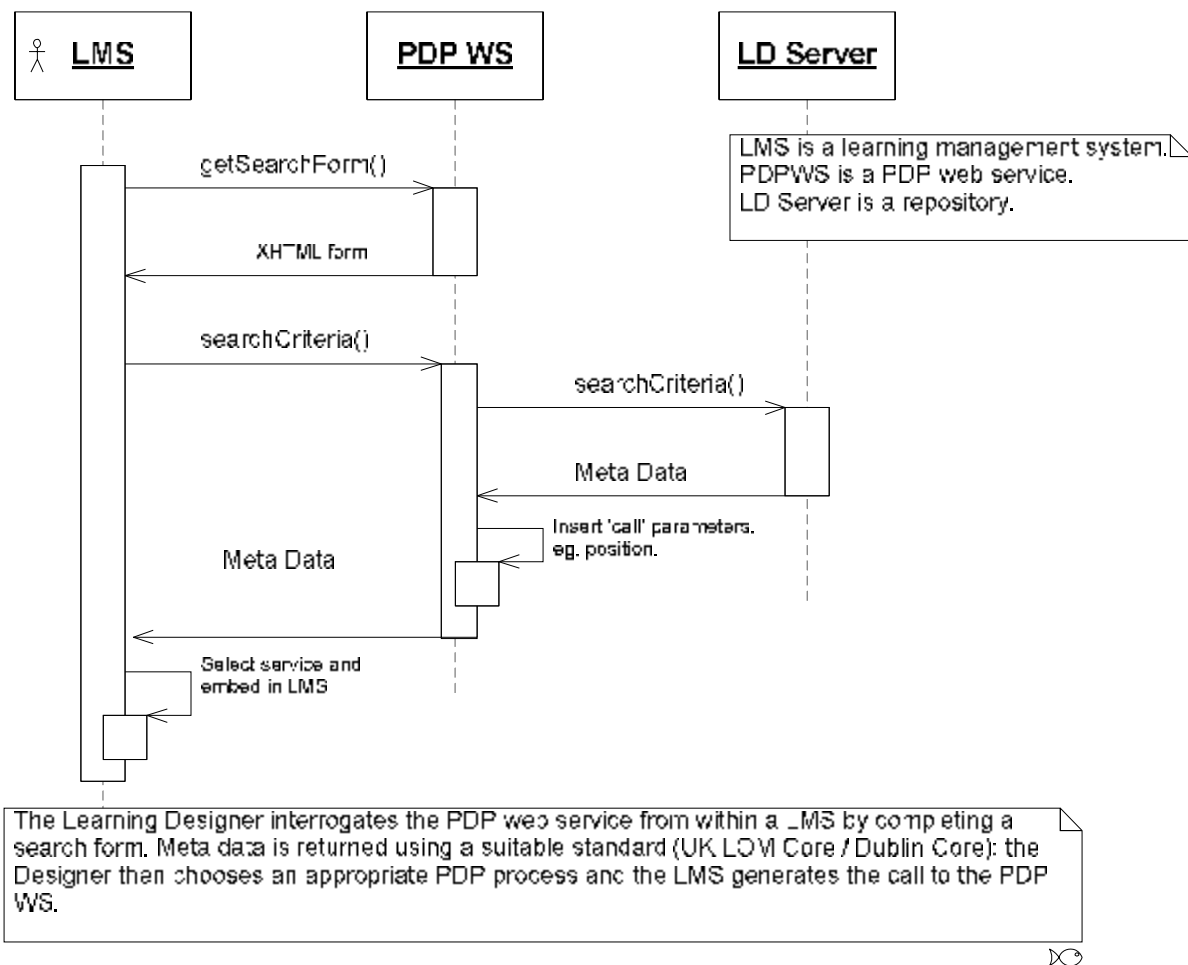
2.4.2.5 PDPWS “search service” (invoked by learning designer)

What is needed is a service to show learning designers the map of all possible “positions” relevant to that PDP service, along with some explanation of what each one is designed to be for. These may be structured hierarchically, or not. The learning designer needs to be able to select any permitted one. The search invocation will be relatively simple: something along the lines of “show me what PDP you offer”.

In contrast to the learner-originated call described later, the designer-originated one will return the complete map of PDP “positions” for that PDPWS, along with metadata describing those positions. As the designers requesting this information are likely to be familiar with LOM, the obvious choice is to treat each PDP position as a learning object, and to base the metadata set to use in this context on the UK LOM Core (<http://www.cetis.ac.uk/profiles/uklomcore>). The UK LOM Core information model, along with many other similar metadata models, has a section (7, “Relation”) dealing with relationships between leaning objects. This can be used to define the hierarchical structure, if appropriate for a particular PDPWS, using the Dublin Core relations “ispartof” and “haspart”.

In summary, we are saying that a PDP system must support some sort of simple search service – the interface for this will be defined as part of the ELF common services layer (see http://www.cetis.ac.uk:8080/frameworks/common_services/search).

The procedure may look something like this:



2.4.2.6 PDPWS “getPDPInteractive” (invoked by learner)

This service is called as the learner interacts with their VLE or other on-line Learning Management System (LMS).

The call may have been placed at a strategic point in the learning environment by the learning designer or tutor. Typically, the learner will click on a link or button to initiate the process; the result will be that control is passed to a learning design player which then will execute the LD that is returned as a response to the service invocation.

2.4.2.6.1 WS request where the learner is identified [4]

This call can be used when the individual learner is registered with the PDPWS. Parameters:

- User ID (from IMS Common Services), see above.
- location of the learners' PIADS, a URL – the WSDL of the PIADS service.
- the ID of a "focal" record, which may be supplied by the PIADS through the PLE; string – Note that this is the value of an XML id= attribute NOT an IMS SourcedId value as the focal record may not be from an IMS compliant record. PIADS must construct and assign unique identifiers to every node it holds.
- set of name-value pairs of other parameters relevant to that PDPWS, for instance: position in PDPWS (default: start); focal time or time span.

The location of the PIADS is required so that a learner can change PIADS instantly, or even run duplicate PIADS. It also could serve as a check on random automated access.

The focal record could be used by any PDPWS but there are no other common parameters. This is why this one has been abstracted from the property list (see below) and is an argument in its own right.

Initially, no focal record or other parameters need to be supplied, and the PDPWS returns a default menu for that learner, as designed and determined by the designers of the PDPWS. As discussed above, that default menu may have one or more of: a search box in which to enter keywords; the top level of a hierarchical menu system; or a fully-displayed structure of nested lists to choose from. Later parameters are provided by the PDPWS itself, during the course of usage.

The last parameter – a set of properties – is a mechanism for the PDP system to be passed implementation-specific information. This list may include the "position" in that PDP system (which may be multi-dimensional); values may be chosen by the learning designer, or filled in by the learner during running of the system. Examples of learner-centred values include focal date and time span – these could be used when planning activities by the use of a Gantt chart – the system needs to know what time frame to use for the chart. Any parameters at all can be passed by this method.

It is tempting to say that specification of the PDP to return (which the learning designer makes) should be separated from any learner supplied values but this would be erroneous. The two sets of parameters are intertwined; for example, the learning designer may ask the user to 'reflect on activities undertaken in academic year 2001-2' or they may ask 'reflect on activities undertaken in an academic year of your choice' and the learner may select 2001-2 as their choice, in both cases the name-value pairs and the LD returned would be identical – it is irrelevant who decided upon the year.

A similar name-value pair approach is used by the OKI's OSIDs – here the caller and the WS must have an understanding of what the parameters are doing – this is a so called 'out of band' agreement (see http://sakaiproject.org/faqs.html#What_are_the_Levels). It is necessary for both the consumer and provider to understand that they need to communicate specific proprietary information in order for the service to work correctly. Even though this does little for interoperability, we feel it is the only solution as all PDP systems will be diverse and require different information to work correctly.

If a “position” is supplied, it would depend entirely on the map of the PDP provided (see 2.5 below). For instance, in LUSID, one possible position could be of analysing the skills used in a particular experience. The corresponding “focal” record would refer to the experience being analysed. In this case, working on one specific record, that record ID can be given. But it is less clear how to specify the records more generally. Typically, working on one record will involve reviewing related records, in a fairly predictable manner. Perhaps ideally, the PDPWS would be able to work out, from the position and the central record, which other records were appropriate – these would in any case be included in the XForms. If they stood to be edited in the activity, they would also appear in the XForms model, presumably as LIP.

One of the functions which should be supported by the PLE is the selection of a record from the PIADS, and the construction of a call to the PDPWS to ask for a menu of PDP activities relating to that record.

WS request where the learner is anonymous [4]

This call is when the learner is not registered with the PDPWS. Parameters:

- XML file giving details of learner and any relevant records, including any accessibility information (e.g. ACCLIP) but not learner ID;
- parameters relevant to the PDPWS (as above: if none are supplied, default to an initial menu);

This second, “anonymous” version of the call poses certain problems which mean that it would only be suitable in a limited set of situations. Firstly, if the learner is not registered with the PDPWS, then the PDPWS may not know where the PIADS for that learner is. When the PDP material is returned, should the information generated or gathered (e.g. on the forms returned) be stored, and if so, where? Conceivably, the material returned by the PDPWS could include a storage “hook”, though this requires greater functionality in the PLE.

Secondly, the decision process, on which the PDPWS bases the decision of exactly what PDP to serve, may be a complex one, which may in some cases require only a little information, and in other cases a lot of information about the learner. Giving the PDPWS the learner ID allows it to initiate the process of gathering the required information about the learner in the most efficient way, from the PIADS. But in the anonymous case, the information which is needed must be pre-supplied. The danger is either supplying too little information, in which case the PDPWS may not return an appropriate chunk of PDP, or too much (say, the complete records of the learner) which might be extremely inefficient and time-consuming.

Thirdly, it is not clear in what format such information should be presented. How would the PDPWS expect the information to be given? We could perhaps argue that it should be in LIP,

on the grounds that all the standard information relevant to an e-portfolio can be given in LIP (see the IMS e-portfolio draft spec, <http://www.imsproject.org/ep/>), but who is to say that no other information can be relevant?

All these reasons add up to the desirability of the learner being identified when using a PDP service. Although anonymous service may be possible, this remains to be investigated in detail.

We do not propose to explore this aspect of the service further here.

2.4.2.6.2 Response Parameter(s) [5]

- IMS Content Package containing IMS LD

The PDPWS returns a piece of LD with embedded web service requests for at least one ‘user input’ form. This LD represents a PDP process to be followed by the learner.

The LD is designed to take the particular learner (category of learner) either through the next decision process, or through the actual PDP activity (the “leaf”). In many cases, the LD component could be a content-free wrapper, meaning “take this form and do it now”. But in other cases, any decision process could for instance be prefaced by a discussion with parent / teacher / tutor / mentor, which can be scripted with the LD. Thus there is no hard-and-fast distinction between navigation and “proper” PDP. Every action can potentially be seen as a process, and it is the job of the LD to coordinate any different actors and scenes in that process.

The LD will be played in the learners LMS; if it is the interactive version, it will at some point invoke the PDP system’s “getInputForm” interface to retrieve a form (represented in XHTML, XForm and IMS LIP) to gather data from the user. This form will use XForm’s ‘model’ capabilities to generate an IMS LIP compliant PDR; the XForm will need to invoke the PIADS “store” web service for the data to be saved.

The navigation through the PDPWS needs also to be seen to. One possibility, if the PDP activity requested is not entirely self-contained, is for the last form supplied (or in cases of navigational pages, possibly the only one) to contain a menu of calls back to the PDPWS with suitable parameters. At present it is difficult to create a proper WS call using SOAP directly from a form or indeed from within IMS LD, so until XForms 1.1 (or a updated IMS LD specification) appears, allowing SOAP submission, the menu could be even as simple as web links, with appropriate query strings, pointing at a program in the PLE (a proxy) which is able to generate the appropriate Web Services call – which may as well be the same system that generates the initial call to the PDPWS.

2.4.2.7 PDPWS “getPDPBatch” (invoked by learner)

This is the service that would be used with a PLE. It is essentially the same as “getPDPInteractive” except that all required elements, including learner data, are contained in the Content Package. The package will include all IMS LD, plus all data entry forms plus the user data relevant to those forms as it was when the package was assembled. As mentioned

elsewhere, the burden of managing the package is placed on the PLE which will also have to ensure that the PIADS are updated when the learner has finished.

2.4.2.7.1 Request Parameter(s)

- See getPDPInteractive above.

2.4.2.7.2 Response Parameter(s)

- See getPDPInteractive above.

2.4.2.8 PDPWS “getInputForm”

This will be invoked from within the LD returned by the “getPDPInteractive” service call. To construct the PDR input form, the PDP system will have to retrieve the form template and then contact the learner’s PIADS to retrieve the appropriate data which will be folded into the form before it is despatched to the user.

In principle, as with any database used for multiple purposes or by more than one person at a time, there has to be some resolution of possible conflict of editing. Possible solutions here are in principle no different from solutions to the same problem in different contexts. However, the facility exists in LIP and possibly in current e-portfolio systems to assign an owner to each piece of information. If this is done carefully, it may be possible to avoid most conflicts by allowing only the owner to edit or delete any particular piece of information. In cases where the ownership of information is actually shared, there will need to be some system to deal with conflicts.

Note that this WS call could effectively log the user into a traditional servlet-based PDP system and assign a sessionID, in other words, LUSID could actually be used in exactly the same way as it currently is (HTML forms plus servlets calls).

2.4.2.8.1 Request Parameter(s)

- User ID (from IMS Common Services);
- location of the learners’ PIADS, a URL – the WSDL of the PIADS service;
- the ID of the form required, as given by the PDPWS in the response to the earlier getPDPInteractive request;
- the ID of a “focal” record, which may be supplied by the PIADS through the PLE;
- set of name-value pairs of other parameters relevant to that PDPWS, for instance: position in PDPWS (default: start); focal time or time span.

2.4.2.8.2 Response Parameter(s)

- XHTML (or the like) file with XForm information plus ‘model’ for data input.

The XForm should hold embedded web service calls to PIADS which it would send standards compliant data to so it can be stored; in the short term this is not possible (as XForms do not support WS calls), so an alternative method could be used.

2.4.3 Generic process for learner-directed use of a PDPWS

2.4.3.1 Batch mode

The learner starts up their LMS (a PLE or a VLE or some other comparable system). The initial menu displays services with which the learner is registered, or freely-available services which the learner has chosen to have on the menu. As registered with the PDPWS, one of the initial menu options is to work with PDPWS. The learner selects that option. An initial call ('getPDPBatch') is made to the PDPWS.

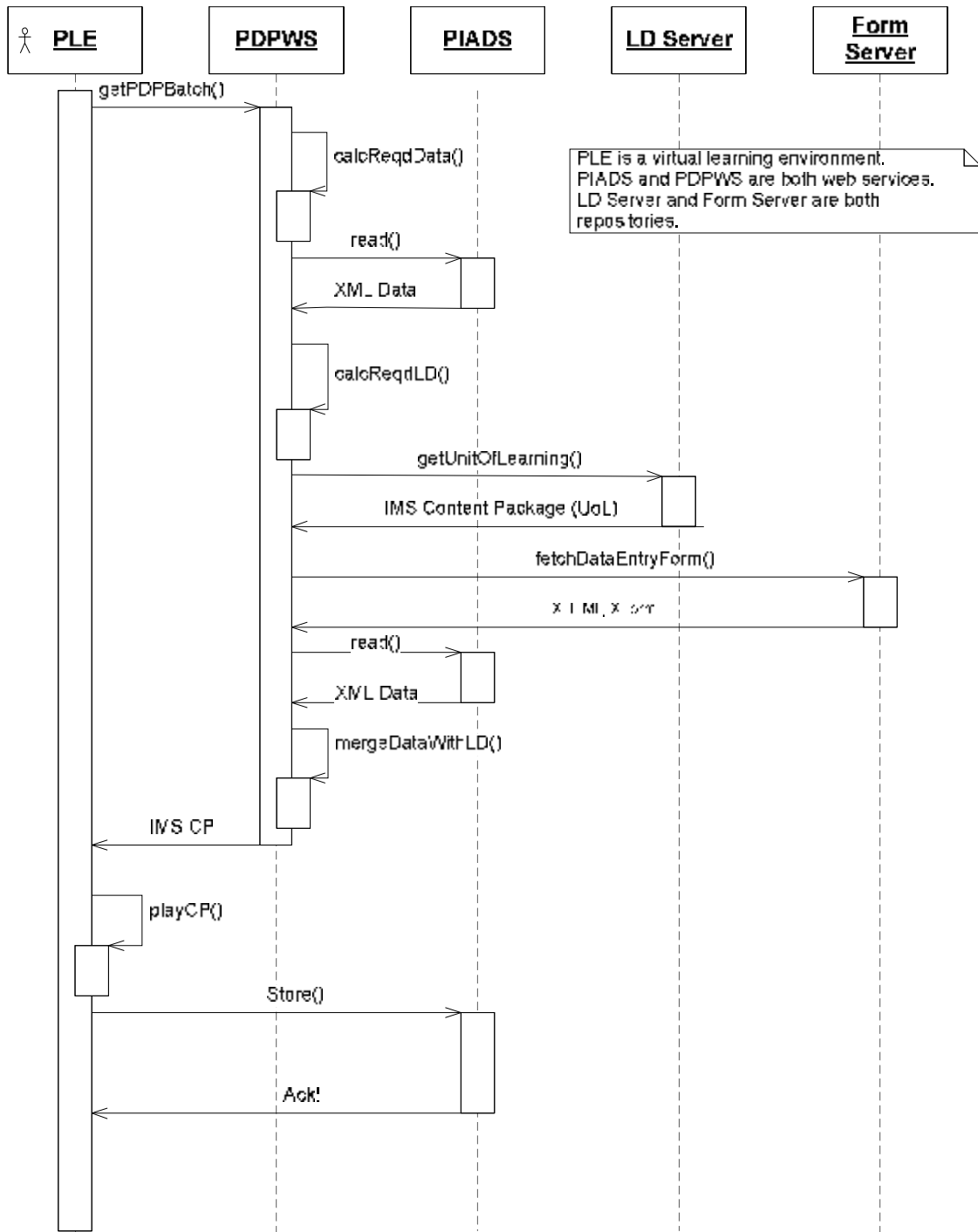
The PDPWS goes through a general process on each call, [4] the initial one included.

1. Check the incoming parameters for consistency, authorisation etc.
2. Get information necessary to find appropriate return:
 - a. check for timeliness of appropriate information;
 - b. consult information available in any PDPWS internal store;
 - c. consult locally available copies of PIADS information;
 - d. consult PIADS;
 - e. if necessary, ask the learner to fill in necessary information where a behaviour is not designed for unspecified information.
3. Determine appropriate material for return.
4. Build up the LD and associated files (with input forms and learner data).
5. Return the file to the LMS. [5]

The LMS then plays any LD sent.

When the LMS reaches the end of the LD to hand, the form(s) need(s) to

6. return any information to the PIADS for storage; [8] (alternatively, this could be done by using PDPWS as a PIADS proxy)
7. either make a new PDPWS call [4] (this process could use a menu as discussed above)
8. or "sign off" – or just stop if there is no explicit signing off



PLE is a virtual learning environment. PIADS and PDPWS are both web services. LD Server and Form Server are both repositories.

The learner will click on a link in the PLE which will initiate the process. The PDP web service is contacted and will work out what PDP should be returned given the learners current situation, i.e. by looking at their records. The LD that is returned should contain all input forms and all learner data. The PLE may now be used without internet access, the PLE must manage the merging of data and the input forms. Finally, once the PLE has on-line access again, the PIADS is contacted to store the data.

2.4.3.2 Interactive mode

The learner starts up their LMS. The initial menu displays services with which the learner is registered, or freely-available services which the learner has chosen to have on the menu. As registered with the PDPWS, one of the initial menu options is to work with PDPWS. The learner selects that option. An initial call ('getPDPInteractive') is made to the PDPWS.

The PDPWS goes through a general process on each call, [4] the initial one included.

1. Check the incoming parameters for consistency, authorisation etc.
2. Get information necessary to find appropriate return:
 - a. check for timeliness of appropriate information;
 - b. consult information available in any PDPWS internal store;
 - c. consult locally available copies of PIADS information;
 - d. consult PIADS;
 - e. if necessary, ask the learner to fill in necessary information where a behaviour is not designed for unspecified information.
3. Determine appropriate material for return.
4. Build up the LD and associated files (with links).
5. Return the file to the LMS. [5]

The LMS then plays any LD sent. Whilst playing,

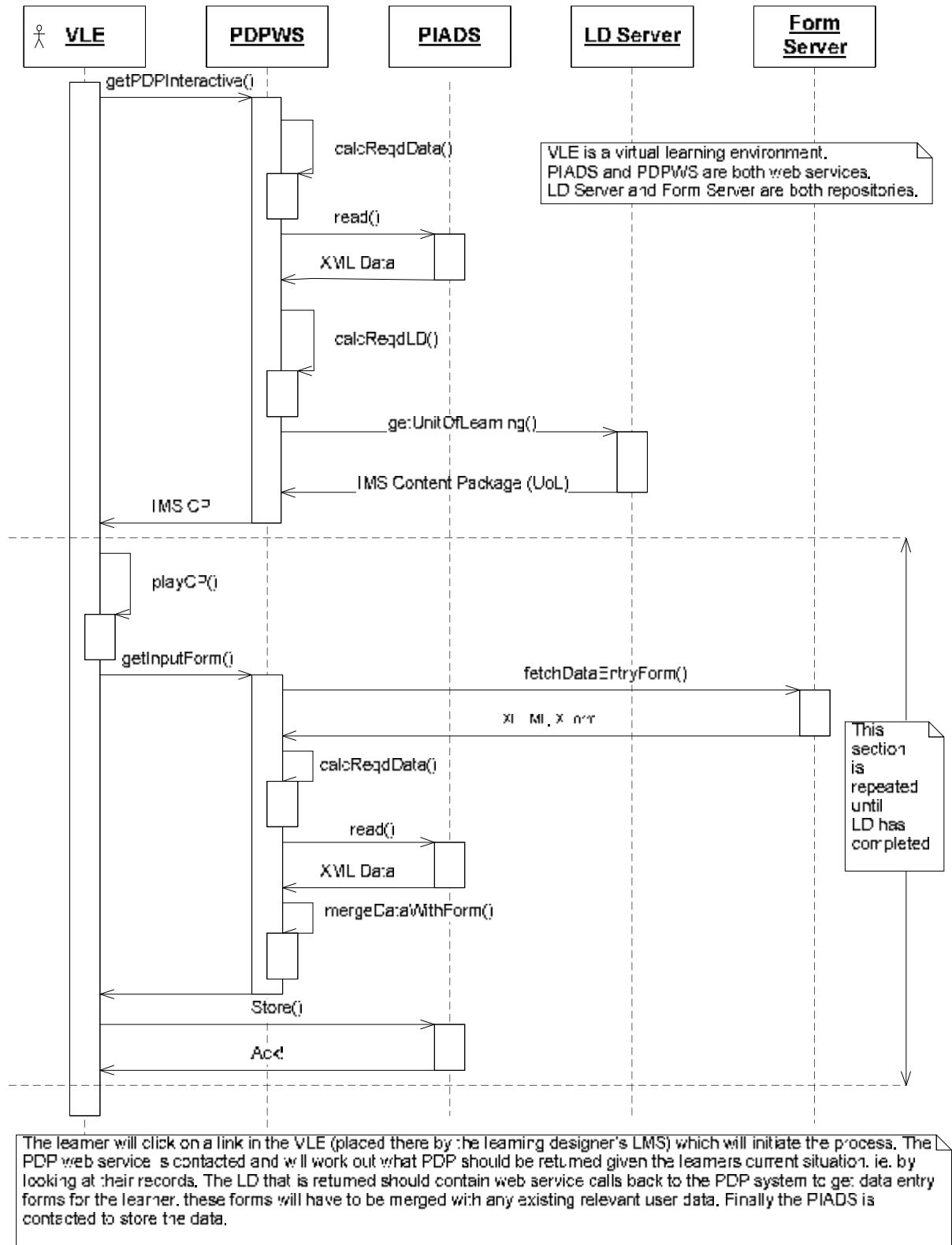
6. the LMS will contact PDPWS to get learner data input form
7. the PDPWS will
 - a. contact the form repository and fetch the form
 - b. contact the learners PIADS and fetch required data
 - c. merge the data with the form and return to LMS

When the learner has completed entering data the LMS will

8. submit the form data to PIADS for storage
9. continue with the next activity as specified by the LD

When the LMS reaches the end of the LD to hand, the form(s) need(s) to

10. either make a new PDPWS call [4] (this process could use a menu as discussed above)
11. or "sign off" – or just stop if there is no explicit signing off



∞

2.5 A few possible nodes in a PDPWS map

Here we could sketch out what happens, in the terms above, for some of the nodes in our model of Personal Theory Building (Appendix A) were it to use such a PDPWS.

2.5.1 Learning log or journal

This refers to the documenting stage in section 1.2.3 above and Appendix A below. This is where a journal or log entry is made about the reactions to the work presentations. In the UML diagram in Appendix A, the steps are (C) – D – G – H.

The learner sets up the learning log just once, not at every interaction. If the log is to be completed regularly, the PLE could prompt the learner. Otherwise, the learner enters spontaneously, through a series of menus, and chooses the log. The PDPWS sends the log form. The learner fills in log form and submits the form. The log information goes to the PIADS. The PDPWS offers some other activity which might be appropriate. If the learner declines, the interaction is then closed.

2.5.2 Review of level of confidence in a skill area

This is represented in the recollection and theorising steps: sections 1.2.4 and 1.2.5 above, and Appendix A below. In the UML diagram, the steps are (K – L) – N.

The learner, as before, starts the PDPWS and goes through menu steps until the skill area is chosen; requests the activity to review level of confidence in a particular skill area, which may be smaller or larger.

The PDPWS sends a form or forms with all the relevant records displayed, and a form for revising the new level of confidence. These may be articulated by the LD, for instance so that a sequence of evidence is gone through in a particular order; or that such a choice is given to the learner.

The learner goes through the LD activity, at the end reaching a form which ends the activity, submits the information to the PIADS, and may select a new position from the PDPWS – for example, the setting goals step which follows here.

2.5.3 Setting goals

This relates to section 1.2.6 above. It is steps P – Q in UML activity diagram of Appendix A.

The learner navigates through to the activity of setting a new goal. The PDPWS returns some LD which specifies that the goal setting must start with a discussion with the learner's mentor. (How this might be implemented is out of scope here.)

After that has finished, the learner fills in a form for the new goal. Submission stores the new goal in the PIADS and sends a message to the mentor about where to find it. The mentor adds a note to the goal. The PDPWS may suggest supplementary questions to the learner.

2.5.4 Discussing an action plan with a learning advisor

This is part of the action planning step in Appendix A and section 1.2.7 above. The UML diagram steps are R – S.

The learner uses the PDPWS (or the PIADS directly) to navigate through records to a particular goal – at that point, the WS call will have the ID of that goal as the focal record. Having identified the goal, the PDPWS could check that the goal and other records were set as visible by the advisor, and then return an activity involve a learning advisor also being sent a form with the goal and plan details, and perhaps opening a collaborative facility such as voice or screen chat at the same time.

2.5.5 Planning an essay

This relates to the embedded PDP scenario already described in some detail in section 2.1.5 above. Let us revisit this as a stand-alone PDP activity. Appendix B attempts to represent this activity in a UML sequence diagram.

The learner starts up PDP from the PLE and navigates through the choices provided by the PDPWS until reaching an option for planning an essay. (We assume this is provided in the PDPWS, though there is no reason why any particular activity should be provided.) At each point, the PDPWS requests any information that is required from the PIADS, in order to properly shape the choice activities (often just menus).

The last choice is of a number of different advanced essay types. The learner chooses a natural science essay. The PDPWS has no more need to consult with the PIADS at this stage (except perhaps to check everything is still in order) and then returns to the PLE a structured natural science essay-planning activity. This has several forms, linked with the LD structure. At the end of the essay planning activity, the output from the forms is collected together and sent as a LIP file to the PIADS for storage.

If the PDPWS is acting in a truly stateless way, there will be no need to inform the PDPWS that the learner has finished with PDP for the moment.

3 Acknowledgements

The authors wish to thank Tunde Varga-Atkins from the University of Liverpool for help with Learning Design; John Harrison from Eidentity Ltd (<http://www.edentity.co.uk>) for useful discussions and comments regarding PIADS and Virtual Home and Bill Olivier from CETIS for useful information regarding IMS Learning Design.

4 References and bibliography for section 1

Anderson, J.R., 1993. *Rules of the Mind*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Bartlett, F.C., 1932. *Remembering*. Cambridge University Press, Cambridge, UK.

- Centre for Recording Achievement, 2004. *Personal Development Planning practice and systems* [online]. Available from: <http://www.cetis.ac.uk/members/PDPcontent>
- Entwistle, N. 2000. *Promoting deep learning through teaching and assessment: conceptual frameworks and educational contents*, (Paper presented at the TLRP Conference, Leicester, November 2000.) Available from: <http://www.ed.ac.uk/etl/docs/entwistle2000.pdf>
- Gough, D.A., Kiwan, D., Sutcliffe, S., Simpson, D. and Houghton, N. (The Evidence for Policy and Practice Information Co-ordinating Centre PDP Review Team) 2003. *A systematic map and synthesis review of the effectiveness of personal development planning for improving student learning* [online]. Available from: http://www.ltsn.ac.uk/application.asp?app=resources.asp&process=full_record§ion=generic&id=231
- Graduate Prospects Ltd, 2004. prospects.ac.uk website.
- Kelly, G.A., 1955. *The Psychology of Personal Constructs*. Norton, New York, USA.
- Perry, W.G., 1999. *Forms of intellectual and ethical development in the college years: a scheme*. Jossey-Bass: John Wiley & Sons, San Francisco, USA. (previously published 1968, 1970).
- Vygotsky, L.S., 1978. *Mind and society: The development of higher mental processes*. Harvard University Press, Cambridge, MA, USA.
- Yancey, K.B., 1998. *Reflection in the Writing Classroom*. Utah State University Press, USA.

Appendix A: activity diagram for personal theory building

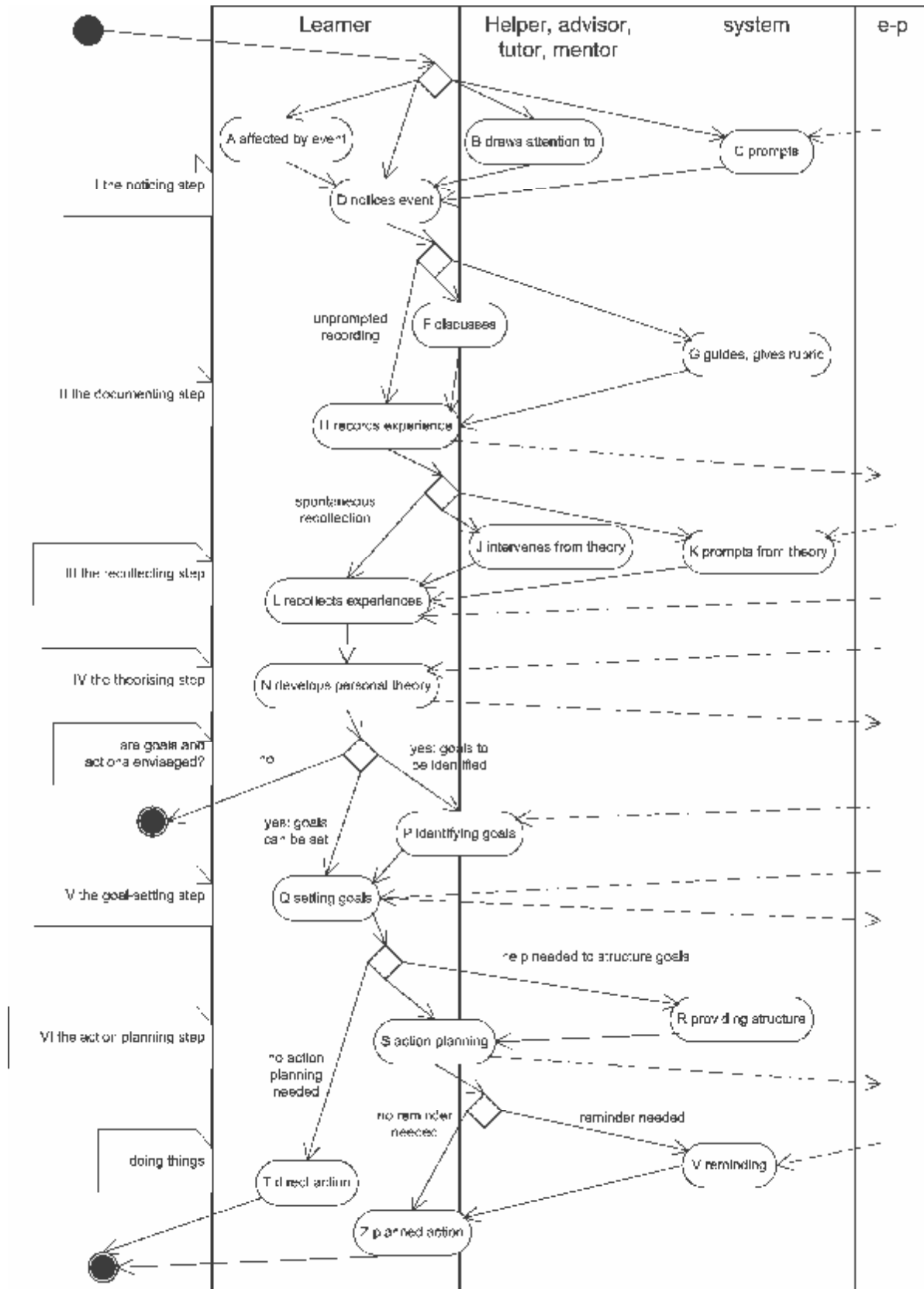


Figure 2: UML Activity Diagram for Personal Theory Building

The diagram in Figure 2 shows four “swim-lanes”. The three main ones correspond to the three types of “actor” named at the top: the learner is the one doing the reflection; the helper is the person helping the learner; and the system is the ICT system used. Within these three columns, if an activity can be shared between the learner and helper, the activity symbol is shown crossing the line. The fourth column labelled “e-p” does not belong to the activity diagram proper, but represents the e-portfolio store, under whatever name, where the personal details of the learner profile are stored. This corresponds to the “PIADS” of Section 2; however, the architecture suggested in Section 2 is not reflected here. The dashed arrows represent possible points at which information could flow to of from the e-portfolio store, to support the activities or resulting from the activities.

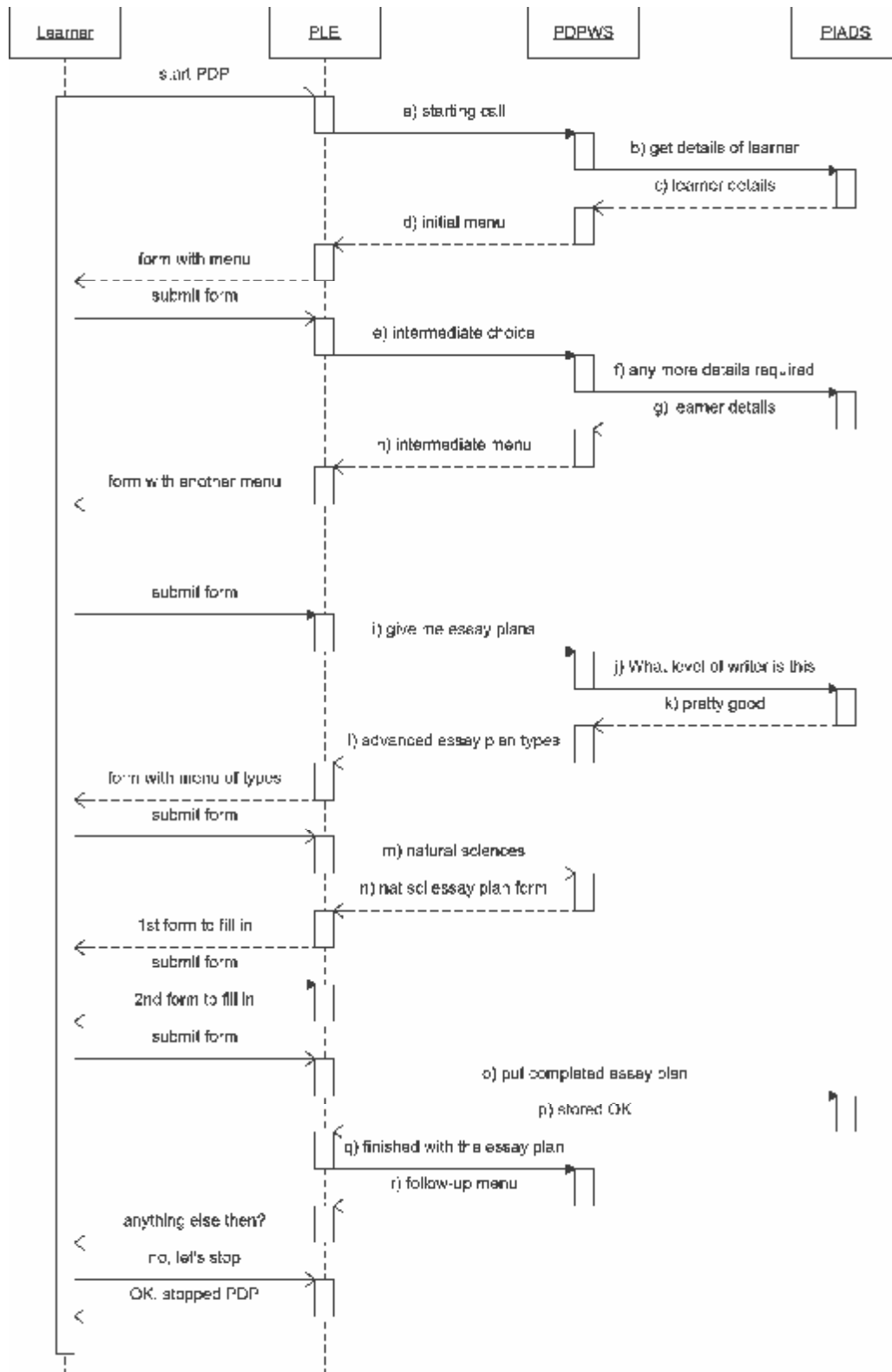
Some of the activities labelled as “A” to “Z” above can also be approximately mapped to the generic activity types which have been listed above, and are numbered below in Appendix C for ease of reference. Other activities, however, have tended to be implicit within PDP and do not find correspondence in the list of generic activities. The table here following indicates some supposed correspondences. However, there is some conceptual distance between the theory-led categories of this model of personal theory building, and the practically-based concepts of the generic activity types, so the correspondence is often rather weak, and there could be arguments for many more associations.

F discusses	7 Reviewing critical incidents
H records experience	1 Compiling list of experiences or past activities, including employment
	30 Writing log (for learning or reflection)
L recollects experiences	3 Reviewing and reflecting on logs
	5 Reviewing experience in response to guidance
	7 Reviewing critical incidents
	14 Assembling evidence for skills
N develops personal theory	9 Relating experiences to skills (or vice versa)
	10 Reviewing / profiling / auditing skills
	10 Reviewing progress in / development of skills
	15 Assessing own learning style

Q setting goals	16 Setting goals for skills development
	17 Setting goals related to subject development
	18 Setting more general personal / social goals
S action planning	20 Originating action plan for the achievement of academic goals
	23 Revising action plan for academic goals in the context of feedback / discussion
	24 Originating action plan for personal / skills development / goals
	25 Revising action plan for personal goals in the context of feedback / discussion
Z planned action	26 Doing exercises alone for skill development
	27 Participating in workshops / classes / sessions for skill development

Table 1: Modelled activities related to generic PDP activities

Appendix B: A model of providing an essay plan by web services



Appendix C: Generic Types

The following lists of generic PDP activities and generic PDP outputs can be found within the <http://www.cetis.ac.uk/members/PDPcontent> website. Items where there are no reports of any usage have been omitted.

Generic activity types

1. Discussing learner's personal situation / experiences
2. Compiling list of experiences or past activities, including employment
3. Reviewing and reflecting on logs
4. Reviewing past written goals and action plans against more recent past experience
5. Reviewing experience in response to guidance
6. Reviewing coursework performance and course experience
7. Reviewing critical incidents
8. Listing achievements / qualifications (with documentation if available)
9. Relating experiences to skills (or vice versa)
10. Reviewing / profiling / auditing skills
11. Reviewing progress in / development of skills
12. Reviewing personal interests
13. Reviewing / reflecting on personal attitudes / values
14. Assembling evidence for skills
15. Assessing own learning style
16. Setting goals for skills development
17. Setting goals related to subject development
18. Setting more general personal / social goals
19. Relating goals to motivations and reasons
20. Originating CV / personal statement / other compilation
21. Revising CV / personal statement / other compilation
22. Originating action plan for the achievement of academic goals
23. Revising action plan for academic goals in the context of feedback / discussion
24. Originating action plan for personal / skills development / goals
25. Revising action plan for personal goals in the context of feedback / discussion
26. Doing exercises alone for skill development
27. Participating in workshops / classes / sessions for skill development
28. Choosing / evaluating suitability of course / module / employment / position
29. Writing application for position / employment / course / programme
30. Writing log (for learning or reflection)
31. Writing individual learning plan
32. Negotiating learning / employment contract

Generic output part types

1. Statement of learner's personal situation
2. Statement / list of past activities / experiences / employment
3. Reflective writing, referring either to experiences or to journals / logs
4. Statement of learning from experience
5. Statement of performance on task or goal
6. Statement / list of formal qualifications
7. Statement / list of other achievements
8. Statement / list of skills
9. Evaluation of skills on a scale
10. Identified strengths and weaknesses in skills
11. Evidence for skills
12. Statement / list of personal interests
13. Statement of learner's attitudes / values
14. Statement of learning preferences
15. Statement of reasons for present module / programme / position
16. Statement of current goals without stated motivation
17. Statement of current goals with stated motivation
18. Statement of motivation / aspirations for future position / employment / course
19. CV / personal statement / other compilation of above items
20. Action plan where learner not directed to particular type of goal
21. Action plan towards next life stage (position, employment or study)
22. Action plan for skills development
23. Action plan for academic /educational / work-related goal(s)
24. Action plan for personal / social goal(s)
25. Learning log or journal intended for later reflection
26. Statement of learner support provided (past) or required (future)
27. Statement of other resources available or required
28. Learning / employment contract (include mention of skills)
29. Individual learning plan
30. Date, location, attendance at event / meeting appointment